NATIONAL TECHNICAL UNIVERSITY OF ATHENS
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING
DIVISION OF INDUSTRIAL ELECTRIC DEVICES AND DECISION SYSTEMS

# Human-in-the-Loop Predictive Analytics for Incident Detection in Smart Transportation Systems

*(Προβλεπτική Αναλυτική Δεδομένων με Ανθρώπινη Παρέμβαση για Ανίχνευση Περιστατικών σε Ευφυή Συστήματα Μεταφορών)*

PhD Thesis

Georgia C. Gkioka

Athens, February 2025

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΩΝ ΔΙΑΤΑΞΕΩΝ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΑΠΟΦΑΣΕΩΝ

# Προβλεπτική Αναλυτική Δεδομένων με Ανθρώπινη Παρέμβαση για Ανίχνευση Περιστατικών σε Ευφυή Συστήματα Μεταφορών

# (Human-in-the-Loop Predictive Analytics for Incident Detection in Smart Transportation Systems)

Διδακτορική Διατριβή

Γεωργία Χ. Γκιόκα

**Τριμελής Συμβουλευτική Επιτροπή:**

*Γρηγόριος Μέντζας, Καθηγητής Ε.Μ.Π. (επιβλέπων)*

*Ιωάννης Ψαρράς, Καθηγητής Ε.Μ.Π.*

*Δημήτριος Ασκούνης , Καθηγητής Ε.Μ.Π.*

*Εγκρίθηκε από την επταμελή εξεταστική επιτροπή την 11η Μαρτίου 2025*

**Επταμελής Εξεταστική Επιτροπή:**

| | | |
|---|---|---|
| *Γρηγόριος Μέντζας*<br>Καθηγητής Ε.Μ.Π.<br>(επιβλέπων) | *Ιωάννης Ψαρράς*<br>Καθηγητής Ε.Μ.Π. | *Δημήτριος Ασκούνης*<br>Καθηγητής Ε.Μ.Π. |
| Grigorios Mentzas Grigorios Mentzas<br>11.03.2025 14:02 | | |
| *Δημήτριος Αποστόλου*<br>Καθηγητής Πανεπιστημίου Πειραιώς | *Ιωάννης Βεργινάδης*<br>Επ. Καθηγητής Ο.Π.Α | *Γεώργιος Τσιχριντζής*<br>Καθηγητής Πανεπιστημίου Πειραιώς |

*Ευάγγελος Σπηλιώτης*
Επ. Καθ. Ε.Μ.Π.

Αθήνα, Μάρτιος 2025

*(Signature)*

.........................................

**Georgia Gkioka**
PhD, Electrical and Computer Engineering National Technical University of Athens

# Περίληψη

Αντικείμενο της διδακτορικής διατριβής είναι η αξιοποίηση της αναλυτικής δεδομένων και της τεχνητής νοημοσύνης στην αυτόματη ανίχνευση περιστατικών στον τομέα των ευφυών συστημάτων μεταφοράς. Η διατριβή προτείνει μια μέθοδο προβλεπτικής αναλυτικής (predictive analytics) η οποία μπορεί να υποστηρίξει την ολοκληρωμένη αυτόματη ανίχνευση περιστατικών, καλύπτοντας όλο τον κύκλο ζωής από τη συλλογή των δεδομένων μέχρι την ανίχνευση σε πραγματικό χρόνο και την επικύρωση από ειδικούς.

Η μέθοδος αυτή συνδυάζει αλγόριθμους τεχνητής νοημοσύνης, προβλεπτικές μεθόδους καθώς και αυτοματοποιημένη μηχανική μάθηση (AutoML) για την πρόβλεψη προγραμματισμένων και μη προγραμματισμένων περιστατικών, ενώ παράλληλα χρησιμοποιεί εργαλεία επεξηγησιμότητας, όπως το LIME και το SHAP, για να εξηγήσει τις αποφάσεις των μοντέλων στους ανθρώπινους χειριστές. Αυτή η επεξηγησιμότητα ενισχύει την εμπιστοσύνη στα συστήματα πρόβλεψης, τα οποία συνήθως αποτελούν “black boxes” και διευκολύνει την κατανόηση των προβλέψεων από μη ειδικούς. Επιπλέον, η έννοια της ανθρώπινης παρέμβασης (human-in-the-loop - HITL) ενσωματώνεται στη διαδικασία ανίχνευσης, επιτρέποντας στους ειδικούς να επιβλέπουν και να διορθώνουν τις αυτόματες προβλέψεις σε πραγματικό χρόνο. Αυτό όχι μόνο βελτιώνει την ακρίβεια των προβλέψεων, αλλά και ενισχύει τη συνεργασία μεταξύ του ανθρώπινου παράγοντα και των αυτόματων συστημάτων. Μάλιστα, αποδεικνύεται μέσω αντίστοιχων πειραμάτων η καθοριστική συμβολή των ειδικών (experts) για τη βελτίωση της απόδοσης του συστήματος με την πάροδο του χρόνου με δυναμικό τρόπο.

Στο πλαίσιο της διατριβής πραγματοποιείται εκτενής βιβλιογραφική μελέτη στα γνωστικά πεδία της προβλεπτικής αναλυτικής δεδομένων, των συστημάτων αυτόματης ανίχνευσης περιστατικών και των τεχνολογιών AutoML, επεξηγησιμότητας και HITL. Αναλύονται οι μεθοδολογίες και τα συστήματα που έχουν αναπτυχθεί μέχρι σήμερα, ενώ στη συνέχεια αναπτύσσεται η προτεινόμενη

μέθοδος. Ακολούθως, αναπτύσσεται πληροφοριακό σύστημα το οποίο δίνει τη δυνατότητα εφαρμογής της προτεινόμενης μεθόδου σε αστικά περιβάλλοντα. Συγκεκριμένα, η μέθοδος εφαρμόζεται σε δύο διαφορετικές πόλεις, την Αθήνα και την Αμβέρσα, για να αξιολογηθεί και να συγκριθεί σε διαφορετικά πλαίσια.

**Keywords:** Τεχνητή Νοημοσύνη, Μηχανική Μάθηση, Αυτόματη Ανίχνευση Περιστατικών, Επεξηγησιμότητα, Έξυπνα Συστήματα Συγκοινωνιών.

# Abstract

The present PhD dissertation explores the need for advanced real-time traffic incident detection systems in urban environments, where the complexity and volume of data often overwhelm traditional methods. The research focuses on integrating advanced data analytics including Machine Learning and Deep Learning, in addition to Automated Machine Learning (AutoML), Human-in-the-Loop (HITL) approaches, and explainability techniques to develop a robust and scalable framework for incident detection. This framework has been tested and validated in real-world scenarios in Athens and Antwerp, where the results demonstrated its effectiveness and superiority compared to traditional methods.

Regarding the structure of the present thesis, in the introduction, the motivation is derived from the growing challenges faced by transportation systems today. Traditional incident detection methods, which rely on predefined rules and manual human monitoring, are increasingly inadequate due to the complex, dynamic nature of modern urban traffic networks. Traffic incidents, whether planned (such as large-scale events or recurring congestion) or unplanned (such as road accidents), require timely detection to prevent traffic and ensure road safety. In order to address this challenge, the research presents a novel framework which combines AI-based techniques that not only automate part of the detection process but also integrate human in the loop to ensure that the system remains adaptable and transparent. The present dissertation explores a detailed literature review which provides a comprehensive overview of the current state of traffic incident detection methods. Moreover, the chapter focuses on the limitations of existing Automatic Incident Detection (AID) systems for identifying both planned and unplanned incidents. These systems, which often employ comparative algorithms or time-series models, tend to be limited in their ability to handle large datasets or complex traffic patterns.

This dissertation aims to overcome the identified challenges by integrating explainability techniques and providing human operators with insight into the

system’s operations. In addressing the research challenges, the dissertation sets out several key questions, beginning with how AI-based systems can be designed for real-time monitoring and prediction of traffic incidents. The methodology combines traditional AI approaches with more advanced techniques, such as AutoML, which automates the model selection and optimization process, and human-in-the-loop (HITL), which ensures that human expertise is taken into consideration in the decision-making process. The combination of these approaches ensures that the system remains flexible and capable of adapting to different urban environments and evolving traffic conditions.

The proposed framework for real-time traffic monitoring and incident detection is thus built upon four key pillars: Data Analytics, Automated Machine Learning, Human-in-the-Loop and Explainability. The first pillar, Data Analytics, focuses on leveraging artificial intelligence to anticipate and detect traffic incidents promptly and efficiently. This involves the use of ML algorithms and data analytics techniques to analyze large amounts of historical and real-time traffic data. Automated Machine Learning, the second pillar, is used to optimize the model-building process, reducing the need for manual intervention and giving the possibility to the system to continuously improve its performance. The third pillar, Explainability, aims to make the AI models used in traffic management understandable and transparent to users. Techniques like SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-agnostic Explanations) are employed to clarify how models produce their predictions. Finally, the fourth pillar, Human-in-the-Loop, ensures that human operators are involved in the decision-making process by reviewing the model’s predictions and by providing continuous feedback, enhancing the system’s reliability and ensuring trust in its outputs. Human input helps correct any errors in the models, ensuring that the system's outputs are realistic. This creates a feedback loop that enhances model performance over time, as operators provide valuable insights and corrections. Lastly, combining HITL and explainability ensures that the incident detection system is not only highly effective but also trusted and accepted by its users.

The dissertation also focuses on the use of AI-driven methodologies for detecting both planned and unplanned traffic incidents. Techniques such as time-series analysis help predict traffic patterns, while machine learning models classify incidents based on historical data. The inclusion of deep learning models, particularly Long Short-Term Memory (LSTM) and Graph Neural Networks (GNNs), improves the system's ability to detect non-recurring incidents by achieving high performance in all established metrics.

These AI models are further enhanced by AutoML, which automates much of the model development process. By using tools like TPOT (Tree-based Pipeline Optimization Tool), the system can automatically select and tune models, reducing the need for manual human intervention in calibrating the employed models and ensuring that the most effective models are deployed for incident detection. Incorporating AutoML into the traffic incident detection framework represents a significant advancement, as it automates the traditionally labor-intensive process of model selection and tuning. This dissertation describes how AutoML was used to optimize machine learning pipelines, improving both the accuracy and efficiency of the system. The AutoML approach was especially valuable in adapting to different urban environments, as demonstrated in the case studies from Athens and Antwerp.

Human oversight plays a crucial role in ensuring the system’s transparency and trustworthiness. By incorporating a Human-in-the-Loop approach, the system allows operators to review and adjust the AI’s predictions in real-time, ensuring that critical decisions are not dependent only to automated processes. Explainability techniques, such as SHAP (SHapley Additive ExPlanations) and LIME (Local Interpretable Model-Agnostic Explanations), are utilized to make the AI-driven system’s predictions more transparent and understandable. These tools help operators understand why certain incidents are flagged, making the system more reliable in high-stakes scenarios such as traffic management during an occurred accident.

The dissertation also presents the technical details of the information system developed for real-time traffic incident detection, named AutoEventX. This system integrates the AI models, AutoML pipelines, and HITL and explainability components into a cohesive architecture capable of processing large volumes of traffic data from multiple sources. The system is designed for scalability, allowing it to handle complex datasets in real-time, and it can be integrated with interactive dashboards that allow traffic managers to monitor conditions, receive incident alerts, understand the reasoning behind the system's predictions and give their input.

To validate the effectiveness of the proposed system, two real-world case studies were conducted in Athens and Antwerp. These case studies provided an opportunity to test the system under different traffic conditions and evaluate its performance in detecting both recurring and non-recurring incidents. For instance, in Athens, the system was used to detect non-recurring incidents, such as accidents or breakdowns, which are common in the city's congested urban environment. The system was able to significantly reduce the number of false positives compared to more traditional methods, providing accurate incident detection in real-time. Moreover, it has been demonstrated that the system is able to detect also recurring congestion problems, in both Athens and Antwerp. The system was able to identify patterns in traffic flow and predict planned incidents, i.e. congestion, before they occurred, allowing traffic managers to potentially take proactive measures to reduce congestion. The results of these case studies demonstrate the system's versatility and effectiveness in managing different types of traffic incidents. The deployment also highlighted the value of the HITL approach, as operators were able to provide real-time feedback that improved the system's accuracy and reliability.

In the conclusion, the dissertation reflects on the significant contributions made to the field of AI-driven traffic incident detection. By combining AutoML, HITL, and explainability techniques, the system presents a major advancement over traditional frameworks and incident detection methods. However, the research also acknowledges several limitations, particularly in terms of data quality and the challenges of handling incomplete or noisy datasets. These limitations point to areas for future research. Future research, in particular, should focus on improved multi-source data integration, incorporating real-time sensor data, social media feeds, weather data and crowdsourced information. Advancements in machine learning, deep learning, and reinforcement learning could optimize model adaptability across different urban environments. Human-AI collaboration could be refined through interactive interfaces, AR/VR tools, and real-time operator feedback to ensure a seamless integration between automated predictions and human decision-making. Enhancing explainability techniques with more advanced tools would also make AI models more transparent, while the expansion of prescriptive analytics would suggest actionable interventions to mitigate congestion. Additionally, incident detection systems should be integrated into broader smart city infrastructures, connecting traffic management with emergency response, and real-time incident management strategies.

To sum up, the contributions of this research include:

1. A novel AI-HITL-AutoML traffic incident detection framework that integrates human oversight with automated ML model optimization.
2. The development of an AutoEventX system, an end-to-end AI-driven platform capable of detecting and predicting both planned and unplanned traffic incidents.
3. The integration of explainability techniques (SHAP, LIME) to enhance model transparency, fostering trust and reliability.
4. The inclusion of human feedback through a dedicated loop, which enables the system to take into account operators' expertise and adapt the outputs accordingly.
5. Real-world validation through case studies in Athens and Antwerp, demonstrating high levels of performance.

The findings of this dissertation contribute to the evolution of intelligent traffic management systems by integrating advanced machine learning techniques with human expertise and explainability approaches. This research not only achieves high performance and levels of transparency in incident detection but also lays the foundation for more adaptive and autonomous data-driven systems in urban mobility.

By putting in place an effective synergy between AI and human operators, this work helps in creating future transportation systems that are more efficient, transparent and safer.

**Keywords:** Artificial Intelligence, Machine Learning, Automatic Incident Detection, Explainability, Smart Transportation Systems.

# Acknowledgments

Completing this doctoral dissertation has been a remarkable journey, one that I could not have undertaken alone. It is with deep gratitude that I acknowledge those who have supported me throughout this endeavor.

First and foremost, I owe immense thanks to my supervisor, Professor G. Mentzas. His guidance, support, and belief in my abilities have been invaluable throughout this research. Beyond his exceptional academic expertise, his kindness and generosity have been a source of inspiration, and I feel privileged to have him as a mentor.

I would also like to extend my appreciation to Professors I. Psarras and D. Askounis, the esteemed members of my advisory committee, as well as to Professors D. Apostolou, I. Verginadis, G. Tzichrintzis, and E. Spiliotis, who participated in the examination of my dissertation. A special thanks to Professor D. Apostolou for his encouragement during my academic journey, especially the last few months.

On a personal note, my family has always been my strong foundation. To my mother, Popi, whose belief in me and my potential motivates me to aim higher; to my father, Christos, whose strength and wisdom continue to inspire me; and to my brother, Antonis, who has always been by my side, supporting me unconditionally — I am profoundly thankful for your love, encouragement, and support.

I am also deeply grateful for the people in my life, friends and companions, who have been there for me through the highs and lows; their unwavering support and understanding have made even the most difficult moments more bearable. To all of you who have been by my side—your presence, kindness, and experiences we shared have helped me grow, both personally and academically, and for that, I am truly thankful.

Last but not least, the collaborative spirit I experienced at the Information Management Unit laboratory played a crucial role in my work and research.

To all who have walked this journey alongside me, thank you for making this achievement not only possible but also profoundly meaningful; it has been a remarkable chapter in my life, one that I will always cherish.

# Table of Contents

# List of Figures

# List of Tables

# Εκτεταμένη Περίληψη

Η παρούσα διδακτορική διατριβή διερευνά την ανάγκη για προηγμένα συστήματα ανίχνευσης κυκλοφοριακών περιστατικών σε πραγματικό χρόνο σε αστικά περιβάλλοντα, όπου η πολυπλοκότητα και ο όγκος των δεδομένων συχνά δυσχεραίνουν την απόδοση των παραδοσιακών μεθόδων. Η έρευνα επικεντρώνεται στην ενσωμάτωση της αυτοματοποιημένης μηχανικής μάθησης (AutoML), των προσεγγίσεων Human-in-the-Loop (HITL) και των τεχνικών επεξηγηματικότητας (Explainability) για την ανάπτυξη ενός καινοτόμου πλαισίου για την ανίχνευση περιστατικών. Αυτό το πλαίσιο (framework) δοκιμάστηκε και επικυρώθηκε σε σενάρια πραγματικού κόσμου στην Αθήνα και την Αμβέρσα, όπου τα αποτελέσματα επιβεβαίωσαν την πρακτικότητα και την αποτελεσματικότητά του.

Στην εισαγωγή, το κίνητρο για τη μελέτη προκύπτει από τις αυξανόμενες προκλήσεις που αντιμετωπίζουν σήμερα τα συστήματα μεταφορών. Οι παραδοσιακές μέθοδοι ανίχνευσης περιστατικών, οι οποίες βασίζονται σε προκαθορισμένους κανόνες και σε μη αυτοματοποιημένη ανθρώπινη παρακολούθηση, είναι όλο και πιο ανεπαρκείς λόγω της πολύπλοκης και δυναμικής φύσης των σύγχρονων αστικών δικτύων κυκλοφορίας. Τα κυκλοφοριακά περιστατικά, είτε είναι προγραμματισμένα (όπως τα έργα οδοποιίας ή η επαναλαμβανόμενη κυκλοφοριακή συμφόρηση) είτε μη προγραμματισμένα (όπως τα ατυχήματα), απαιτούν έγκαιρη ανίχνευση για την πρόληψη της συμφόρησης και τη διασφάλιση της οδικής ασφάλειας. Αντιμετωπίζοντας αυτή την πρόκληση, η παρούσα διδακτορική διατριβή προτείνει και παρουσιάζει έναν καινοτόμο συνδυασμό τεχνικών βασισμένων στην Τεχνητή Νοημοσύνη (ΤΝ) που όχι μόνο αυτοματοποιούν τμήματα της διαδικασίας ανίχνευσης, αλλά και ενσωματώνουν την ανθρώπινη επίβλεψη και την επεξηγηματικότητα για να διασφαλίσουν ότι το σύστημα παραμένει προσαρμόσιμο και διαφανές.

Η βιβλιογραφική ανασκόπηση που πραγματοποιήθηκε κατά τη διάρκεια συγγραφής της παρούσας διατριβής παρέχει μια ολοκληρωμένη επισκόπηση της τρέχουσας κατάστασης των μεθόδων ανίχνευσης κυκλοφοριακών περιστατικών. Τα περιστατικά ή συμβάντα αναφέρονται σε «κάθε μη επαναλαμβανόμενο γεγονός που προκαλεί μείωση της χωρητικότητας των οδών ή μη φυσιολογική αύξηση της ζήτησης» . Τα συμβάντα μπορούν να ταξινομηθούν ως προγραμματισμένα ή μη προγραμματισμένα γεγονότα, όπως φαίνεται στο Σχήμα 0-1. (Nikolaev, Sapego, Ivakhnenko, Mel'nikova, & Stroganov, 2017)


**Figure 0-1: Ταξινόμηση περιστατικών. (Nikolaev, Sapego, Ivakhnenko, Mel'nikova, & Stroganov, 2017)**

Στο πλαίσιο της διεξαχθείσας βιβλιογραφικής ανασκόπησης, οι αλγόριθμοι για την αυτόματη ανίχνευση περιστατικών ομαδοποιούνται σε τρεις ευρείες κατηγορίες, ως συγκριτικοί, χρονοσειρές και αλγορίθμους τεχνητής νοημοσύνης (στατιστικοί, μηχανικής μάθησης και βαθιάς μάθησης), όπως φαίνεται στο Σχήμα 0-2.

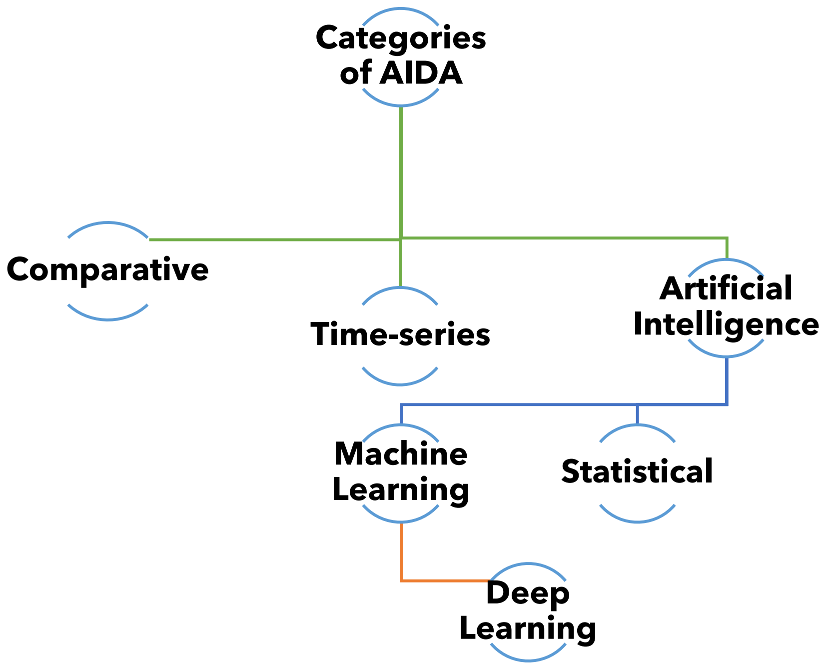



**Figure 0-2: Ταξινόμηση αλγορίθμων αυτόματης ανίχνευσης περιστατικών.**

Επιπλέον, το κεφάλαιο που αφορά τη βιβλιογραφική επισκόπηση αναφέρεται και στους περιορισμούς των υφιστάμενων συστημάτων αυτόματης ανίχνευσης για τον εντοπισμό προγραμματισμένων και μη προγραμματισμένων περιστατικών. Αυτά τα συστήματα, τα οποία συχνά χρησιμοποιούν συγκριτικούς αλγορίθμους ή μοντέλα χρονοσειρών, τείνουν να είναι περιορισμένα ως προς την ικανότητά τους να χειρίζονται μεγάλα σύνολα δεδομένων ή πολύπλοκα πρότυπα κυκλοφορίας. Οι εξελίξεις στη μηχανική μάθηση (ML) και, πιο πρόσφατα, στην αυτόματη μηχανική μάθηση (AutoML) έχουν επιτρέψει μεγαλύτερη ακρίβεια και αυτοματοποίηση στον εντοπισμό περιστατικών. Στο Σχήμα 0-3, παρουσιάζεται λεπτομερώς η ταξινόμηση των τεχνικών ML που χρησιμοποιούνται στην αυτοματοποιημένη ανίχνευση περιστατικών κυκλοφορίας.

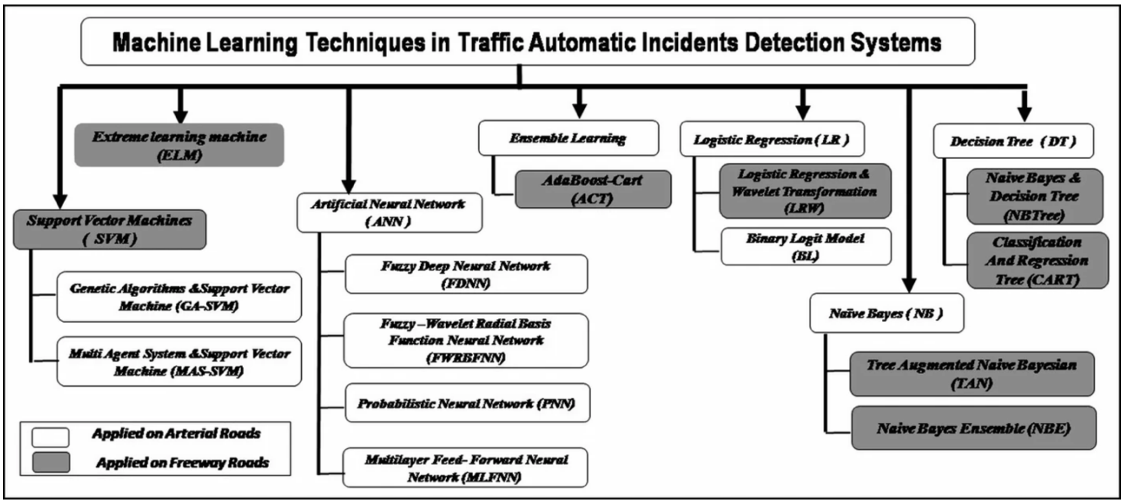



**Figure 0-3: Ταξινόμηση των τεχνικών μηχανικής μάθησης που χρησιμοποιούνται στην αυτόματη ανίχνευση περιστατικών κυκλοφορίας. (Hireche & Dennai, 2020)**

Ωστόσο, αυτά τα συστήματα ML και AutoML λειτουργούν συχνά ως «μαύρα κουτιά», καθιστώντας τις προβλέψεις τους δύσκολες στην ερμηνεία και περιορίζοντας την εμπιστοσύνη των χρηστών. Ένα άλλο κρίσιμο ζήτημα είναι οι υπάρχουσες χρησιμοποιούμενες τεχνικές για τον καθαρισμό των δεδομένων, όπου οι ανακρίβειες στα δεδομένα κίνησης, όπως τα σφάλματα ή οι χαμηλοί ρυθμοί δειγματοληψίας, μπορούν να εισάγουν κενά στην ανίχνευση της κίνησης των οχημάτων και τα κενά αυτά δημιουργούν αβεβαιότητα στην ακριβή παρακολούθηση των περιστατικών. Επιπλέον, ο χειρισμός της χωροχρονικής πολυπλοκότητας των δεδομένων κινητικότητας, όπου τόσο η θέση όσο και ο χρόνος αποτελούν βασικούς παράγοντες, αποτελεί ένα ακόμη τεχνικό εμπόδιο, καθώς τα παραδοσιακά μοντέλα δυσκολεύονται να ενσωματώσουν αυτά τα δεδομένα σε πραγματικό χρόνο. Τέλος, ενώ τα αυτοματοποιημένα συστήματα είναι επιθυμητά, η ενσωμάτωση της ανθρώπινης εμπειρογνωμοσύνης στις διαδικασίες λήψης αποφάσεων σε πραγματικό χρόνο παραμένει μια ανοιχτή πρόκληση, με τα τρέχοντα συστήματα να μη λαμβάνουν συχνά υπόψιν τον ρόλο της ανθρώπινης παρέμβασης σε κατάστασης διαχείρισης κρίσεων, όπως κατά τη διάρκεια ενός ατυχήματος.

Για την αντιμετώπιση αυτών των προκλήσεων, με βάση πολλές σχετικές εργασίες και ερευνητικές δημοσιεύσεις που έχουν αξιολογηθεί, προτείνεται ότι η μελλοντική έρευνα θα πρέπει να επικεντρωθεί σε διάφορους τομείς όπως η ποιότητα των δεδομένων, η διαδικασία συλλογής τους και η επεξηγησιμότητα των συστημάτων και των προβλέψεών τους. Πιο συγκεκριμένα, βελτιωμένες τεχνικές καθαρισμού δεδομένων είναι απαραίτητες για τη διαχείριση του αυξανόμενου όγκου δεδομένων κινητικότητας και τη διασφάλιση υψηλής ποιότητας δεδομένων. Η μεροληψία κατά τη συλλογή δεδομένων μπορεί να μετριαστεί με την ανάπτυξη πιο περιεκτικών μεθόδων που λαμβάνουν υπόψη όλους τους τρόπους μεταφοράς και τις κοινωνικοοικονομικές ομάδες. Η βελτίωση της επεξηγηματικότητας των μοντέλων ML θα αυξήσει την αξιοπιστία τους, ιδίως σε πλαίσια που αφορούν πολλούς ενδιαφερόμενους, όπως οι city planners ή οι transport operators. Θα πρέπει να δοθεί προτεραιότητα σε εξειδικευμένα χωροχρονικά μοντέλα που χειρίζονται εγγενώς τόσο δεδομένα με βάση τον χρόνο όσο και δεδομένα με βάση τη θέση για τη βελτίωση της ακρίβειας ανίχνευσης. Τέλος, η ανάπτυξη υβριδικών συστημάτων Human-in-the-Loop που συνδυάζουν αυτοματοποιημένες διαδικασίες ML με την ανθρώπινη εμπειρία θα συμβάλει στη διασφάλιση πιο αξιόπιστης ανίχνευσης περιστατικών.

Η παρούσα διατριβή στοχεύει να ξεπεράσει τις προκλήσεις που εντοπίστηκαν ενσωματώνοντας τεχνικές επεξηγηματικότητας και παρέχοντας στους ανθρώπινους χειριστές περισσότερο έλεγχο και εικόνα των λειτουργιών του συστήματος. Για την αντιμετώπιση των ερευνητικών προκλήσεων, η διατριβή θέτει διάφορα ερωτήματα, ξεκινώντας από τον τρόπο με τον οποίο μπορούν να σχεδιαστούν συστήματα βασισμένα στην τεχνητή νοημοσύνη για παρακολούθηση και πρόβλεψη κυκλοφοριακών περιστατικών σε πραγματικό χρόνο. Η μεθοδολογία συνδυάζει παραδοσιακές προσεγγίσεις ML με πιο προηγμένες τεχνικές, όπως AutoML, η οποία αυτοματοποιεί τη διαδικασία επιλογής και βελτιστοποίησης μοντέλων, και τεχνικές HITL, με στόχο τη διασφάλιση πως η ανθρώπινη εμπειρογνωμοσύνη συμμετέχει σε βασικά σημεία λήψης αποφάσεων. Ο συνδυασμός αυτών των προσεγγίσεων διασφαλίζει ότι το σύστημα παραμένει ευέλικτο και ικανό να προσαρμόζεται σε διαφορετικά αστικά περιβάλλοντα και εξελισσόμενες συνθήκες κυκλοφορίας.

Στον συνοπτικό Πίνακα 0-1 αναφέρονται οι ερευνητικές προκλήσεις στις οποίες προσπαθεί να απαντήσει η διδακτορική διατριβή, καθώς και οι αντίστοιχες παράμετροι από τις οποίες αποτελούνται.

**Table 0-1: Ερευνητικά ερωτήματα και οι αντίστοιχες παράμετροι.**

| Ερευνητικά ερωτήματα | Παράμετροι |
| --- | --- |
| Ποια είναι τα βασικά στοιχεία και οι μεθοδολογίες για την παρακολούθηση και την πρόβλεψη σε πραγματικό χρόνο για την έγκαιρη ανίχνευση κυκλοφοριακών περιστατικών με βάση την τεχνητή νοημοσύνη; | • Ποια είναι τα χαρακτηριστικά της κυκλοφορίας σε περίπτωση κυκλοφοριακών περιστατικών;<br>• Ποιες είναι οι βασικές πηγές δεδομένων για την παρακολούθηση της κυκλοφορίας σε πραγματικό χρόνο και την ανίχνευση περιστατικών;<br>• Ποιες είναι οι κύριες κατηγορίες αλγορίθμων για την ανίχνευση περιστατικών; Ποια είναι τα δυνατά και αδύναμα σημεία κάθε κατηγορίας;<br>• Ποιοι αλγόριθμοι τεχνητής νοημοσύνης είναι πιο αποτελεσματικοί και έχουν προταθεί διεξοδικά από τη βιβλιογραφία για την ανίχνευση κυκλοφοριακών περιστατικών; Ποια είναι τα πλεονεκτήματα και οι περιορισμοί του καθενός;<br>• Ποιες μετρικές απόδοσης επιλέγονται για την αξιολόγηση της αποτελεσματικότητας των συστημάτων ανίχνευσης κυκλοφοριακών περιστατικών; |
| Πώς μπορούν να αξιοποιηθούν οι ανθρωποκεντρικές παραδοσιακές και αυτοματοποιημένες τεχνολογίες ML για την | • Ποια είναι τα βήματα για τη δημιουργία ενός ολοκληρωμένου πλαισίου/μεθοδολογίας με χρήση ΤΝ για την ανίχνευση προγραμματισμένων και μη προγραμματισμένων περιστατικών σε πραγματικό χρόνο;<br>• Πώς μπορεί να ενσωματωθεί η ανθρώπινη εμπειρογνωμοσύνη και παρέμβαση σε |

| | |
|---|---|
| ανάπτυξη ενός ολοκληρωμένου πλαισίου για την ανίχνευση κυκλοφοριακών περιστατικών σε πραγματικό χρόνο και την παρακολούθηση αστικών δικτύων μεταφορών; | συστήματα παρακολούθησης της κυκλοφορίας και ανίχνευσης περιστατικών με βάση την ΤΝ; |
| Πώς οι μεθοδολογίες και οι αλγόριθμοι που βασίζονται στην ΤΝ ενισχύουν την ανίχνευση προγραμματισμένων και μη προγραμματισμένων περιστατικών κυκλοφορίας; | • Ποια είναι τα κύρια πλεονεκτήματα της χρήσης ΤΝ για την ανίχνευση κυκλοφοριακών περιστατικών σε σύγκριση με τις παραδοσιακές μεθόδους;<br>• Ποιοι περιορισμοί ή προκλήσεις παραμένουν στις τρέχουσες προσεγγίσεις που βασίζονται στην ΤΝ;<br>• Πώς αποδίδουν τα διάφορα μοντέλα μηχανικής μάθησης και βαθιάς μάθησης στο πλαίσιο της ανίχνευσης κυκλοφοριακών περιστατικών; Υπάρχουν διαφορές μεταξύ της ανίχνευσης προγραμματισμένων και μη προγραμματισμένων περιστατικών;<br>• Ποια είναι τα βασικά χαρακτηριστικά και οι παράμετροι που επηρεάζουν την αποτελεσματικότητα των μοντέλων τεχνητής νοημοσύνης; |
| Πώς μπορούν οι τεχνικές AutoML να ενισχύσουν την ανάπτυξη μοντέλων ΤΝ για την ανίχνευση | • Τι είναι η αυτοματοποιημένη μηχανική μάθηση και ποιος ο ρόλος της στο πλαίσιο των ευφυών συστημάτων μεταφορών; |

| | |
|---|---|
| κυκλοφοριακών περιστατικών; | • Ποιες τεχνικές ή εργαλεία AutoML είναι καταλληλότερα για την ανίχνευση κυκλοφοριακών περιστατικών;<br>• Πώς μπορεί να αυτοματοποιηθεί αποτελεσματικά η επιλογή και η βελτιστοποίηση μοντέλων;<br>• Πώς συγκρίνονται οι τεχνικές AutoML με τις παραδοσιακές μεθόδους όσον αφορά την απόδοση; |
| Πώς διασφαλίζεται ότι οι προβλέψεις των συστημάτων ανίχνευσης κυκλοφοριακών περιστατικών που βασίζονται στην ΤΝ είναι επεξηγήσιμες και αξιόπιστες, και πώς μπορεί να ενσωματωθεί η ανατροφοδότηση από εμπειρογνώμονες; | • Ποιος είναι ο ρόλος της ανθρώπινης ανατροφοδότησης και πώς μπορεί να αξιοποιηθεί σε συστήματα βασισμένα στην ΤΝ;<br>• Ποιες τεχνικές επεξηγηματικότητας μπορούν να χρησιμοποιηθούν για να καταστούν οι προβλέψεις της ΤΝ κατανοητές και ποια εργαλεία μπορούν να ενσωματωθούν σε τέτοια συστήματα;<br>• Ποιοι μηχανισμοί μπορούν να χρησιμοποιηθούν για την ενσωμάτωση της ανατροφοδότησης εμπειρογνωμόνων με τη χρήση μιας προσέγγισης «Human in the Loop» σε ευφυή συστήματα μεταφορών που βασίζονται στην ΤΝ, ώστε να βελτιωθεί η ποιότητα των προβλέψεων;<br>• Ποια είναι τα αποτελέσματα της ενσωμάτωσης της ανθρώπινης παρέμβασης όσον αφορά την απόδοση κατά την επανεκπαίδευση των μοντέλων ML; Οι τεχνικές επεξηγηματικότητας |

| | |
|---|---|
| | έχουν αντίκτυπο στην αξιοπιστία του συστήματος; |

Το προτεινόμενο πλαίσιο για την παρακολούθηση της κυκλοφορίας σε πραγματικό χρόνο και την ανίχνευση περιστατικών βασίζεται σε τέσσερις βασικούς πυλώνες: αναλυτική δεδομένων, αυτοματοποιημένη μηχανική μάθηση, ανθρώπινη παρέμβαση (Human-in-the-Loop) και επεξηγηματικότητα, όπως απεικονίζεται στο Σχήμα 0-4. Ο πρώτος πυλώνας, η ανάλυση δεδομένων, επικεντρώνεται στην αξιοποίηση της τεχνητής νοημοσύνης για την πρόβλεψη και τον άμεσο εντοπισμό περιστατικών κυκλοφορίας. Αυτό περιλαμβάνει τη χρήση εξελιγμένων αλγορίθμων για την ανάλυση μεγάλου όγκου ιστορικών δεδομένων καθώς και δεδομένων κυκλοφορίας σε πραγματικό χρόνο. Εντοπίζοντας μοτίβα και τάσεις, το σύστημα μπορεί να προβλέπει πιθανά συμβάντα και να παρέχει έγκαιρες προειδοποιήσεις. Η αυτοματοποιημένη μηχανική μάθηση, ο δεύτερος πυλώνας, χρησιμοποιείται για τη βελτιστοποίηση της διαδικασίας δημιουργίας και προσαρμογής μοντέλων, μειώνοντας την ανάγκη για χειροκίνητη παρέμβαση και επιτρέποντας στο σύστημα να βελτιώνει συνεχώς τις επιδόσεις του. Ο τρίτος πυλώνας, η Επεξηγησιμότητα, ασχολείται με το να καταστούν οι προβλέψεις των μοντέλων τεχνητής νοημοσύνης που χρησιμοποιούνται στη διαχείριση της κυκλοφορίας κατανοητές και διαφανείς στους χρήστες. Τεχνικές όπως οι SHAP (SHapley Additive exPlanations) και LIME (Local Interpretable Model-agnostic Explanations) χρησιμοποιούνται για να αποσαφηνίσουν τον τρόπο με τον οποίο τα μοντέλα καταλήγουν στις προβλέψεις τους. Τέλος, ο τέταρτος πυλώνας, Human-in-the-Loop, διασφαλίζει ότι οι ανθρώπινοι χειριστές συμμετέχουν στη διαδικασία λήψης αποφάσεων, επανεξετάζοντας τις προβλέψεις του μοντέλου και παρέχοντας συνεχή ανατροφοδότηση, ενισχύοντας την αξιοπιστία του συστήματος και εξασφαλίζοντας εμπιστοσύνη στα αποτελέσματά του. Η ανθρώπινη συμβολή συμβάλλει στη διόρθωση τυχόν σφαλμάτων στο μοντέλο, διασφαλίζοντας ότι οι εκροές (outputs) του συστήματος είναι ρεαλιστικές και ανταποκρίνονται στις πραγματικές συνθήκες. Αυτό δημιουργεί έναν βρόχο ανατροφοδότησης που βελτιώνει την απόδοση του επιλεγμένου μοντέλου με την πάροδο του χρόνου, καθώς οι χειριστές παρέχουν πολύτιμες πληροφορίες και διορθώσεις. Τέλος, ο συνδυασμός του HITL και της επεξηγηματικότητας εξασφαλίζει ότι το σύστημα ανίχνευσης περιστατικών δεν είναι μόνο αποτελεσματικό αλλά και αξιόπιστο και αποδεκτό από τους χρήστες του.

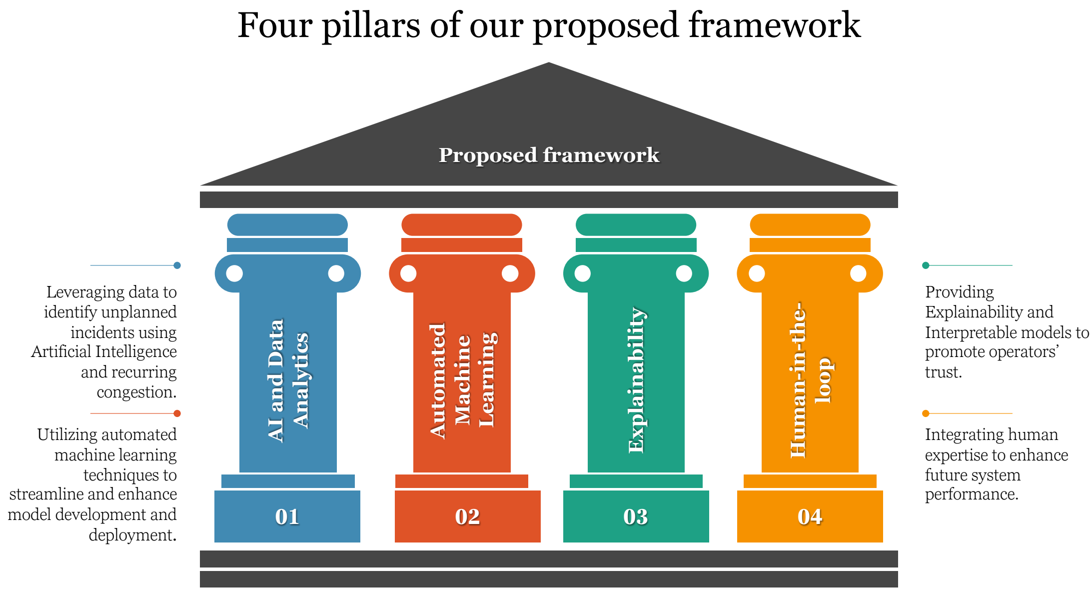



**Figure 0-4: Οι τέσσερις πυλώνες του προτεινόμενου πλαισίου.**

Ο στόχος της προτεινόμενης προσέγγισής μας είναι να βελτιώσουμε τα συστήματα διαχείρισης της κυκλοφορίας αξιοποιώντας προηγμένες τεχνικές ανάλυσης και τεχνητής νοημοσύνης μαζί με την αντιμετώπιση των ερευνητικών προκλήσεων που αναφέρθηκαν παραπάνω, ενσωματώνοντας την επεξηγηματικότητα και τις προσεγγίσεις human-in-the-loop. Η προτεινόμενη μεθοδολογία και τα επιμέρους βήματα απεικονίζονται στο Σχήμα 0-5. Η διαδικασία είναι πολύπλοκη και περιλαμβάνει διάφορες φάσεις, καθεμία από τις οποίες είναι κρίσιμη για τη συνολική αποτελεσματικότητα του συστήματος όπως φαίνεται και στο ακόλουθο σχήμα (Σχήμα 0-5).

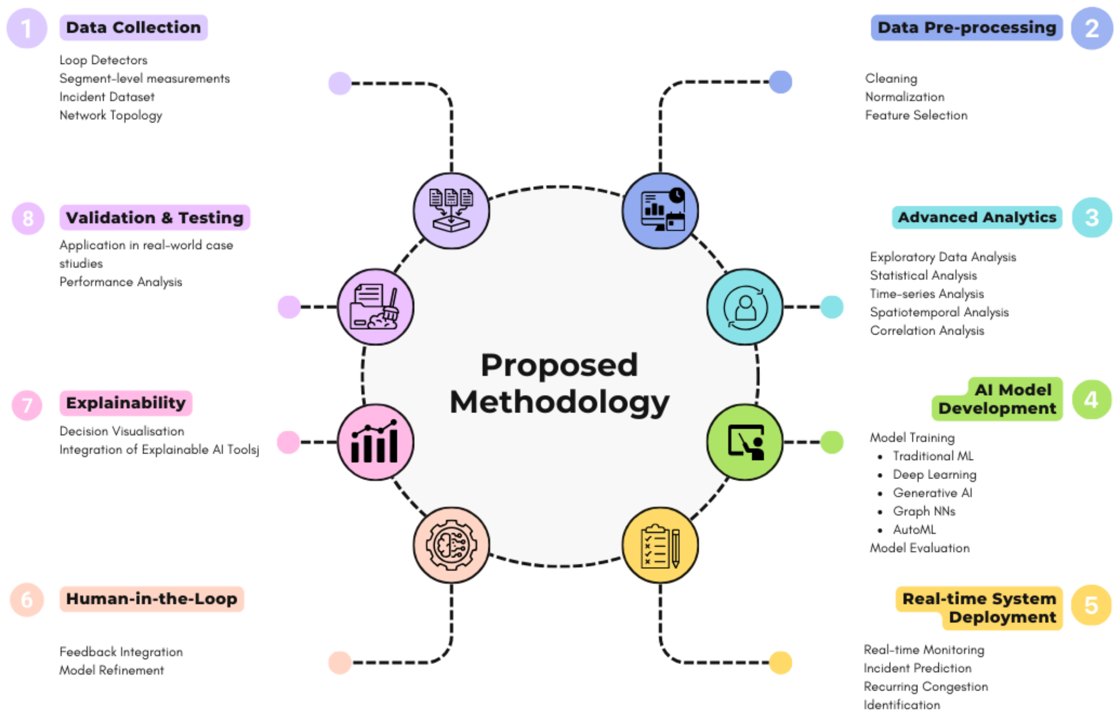



**Figure 0-5: Η προτεινόμενη μεθοδολογία μας.**

Η παρούσα διατριβή αποσκοπεί στην παροχή μιας ολοκληρωμένης επισκόπησης της εφαρμογής μεθοδολογιών που βασίζονται στην ΤΝ στον τομέα της ανίχνευσης κυκλοφοριακών περιστατικών, με ιδιαίτερη έμφαση τόσο σε προγραμματισμένα όσο και σε μη προγραμματισμένα περιστατικά. Παρουσιάζεται η προτεινόμενη προσέγγιση βασισμένη στην ΤΝ για την ανίχνευση κυκλοφοριακών περιστατικών με λεπτομερή περιγραφή των θεμελιωδών τεχνικών που χρησιμοποιούνται. Αυτό περιλαμβάνει μια σε βάθος ανάλυση του τρόπου με τον οποίο εφαρμόζονται μοντέλα ΤΝ, ιδίως αλγόριθμοι μηχανικής μάθησης και βαθιάς μάθησης, για την ανίχνευση περιστατικών. Ιδιαίτερη προσοχή δίνεται στη διάκριση μεταξύ των διαδικασιών ανίχνευσης προγραμματισμένων περιστατικών, όπως η επαναλαμβανόμενη κυκλοφοριακή συμφόρηση, και μη προγραμματισμένων περιστατικών, όπως τα ατυχήματα ή το ξαφνικό κλείσιμο δρόμων.

Ένα από τα κύρια κίνητρα για την εφαρμογή της ΤΝ στην ανίχνευση κυκλοφοριακών περιστατικών είναι η αποτελεσματικότητα της διαχείρισης μεγάλου όγκου δεδομένων και άρα η βελτίωση των δυνατοτήτων ανίχνευσης. Σε αντίθεση με τις παραδοσιακές μεθόδους, οι μεθοδολογίες ΤΝ μπορούν να αναλύουν μεγάλα σύνολα δεδομένων συνεχώς και σε πραγματικό χρόνο. Αυτό επιτρέπει τον εντοπισμό μοτίβων κυκλοφορίας και ανωμαλιών που υποδεικνύουν περιστατικά πολύ πιο γρήγορα και με μεγαλύτερη ακρίβεια από ό,τι οι ανθρώπινοι χειριστές. Ένα άλλο βασικό κίνητρο είναι η ικανότητα των συστημάτων ΤΝ να διευκολύνουν την παρακολούθηση και την ανταπόκριση σε πραγματικό χρόνο. Η γρήγορη επεξεργασία των δεδομένων και η δυνατότητα λήψης αποφάσεων βάσει δεδομένων σε πραγματικό χρόνο μπορούν να μειώσουν σημαντικά τον χρόνο απόκρισης σε περιστατικά κυκλοφορίας.

Με βάση τη φύση του περιστατικού, δηλαδή αν είναι προγραμματισμένο ή μη, διερευνώνται διαφορετικές τεχνικές. Για μη προγραμματισμένα περιστατικά επιλέγονται ορισμένοι από τους πιο ευρέως χρησιμοποιούμενους αλγόριθμους και μεθόδους τελευταίας τεχνολογίας. Για το λόγο αυτό, επιλέχθηκε να μην δοθεί έμφαση σε συγκριτικούς αλγορίθμους ή αλγορίθμους χρονοσειρών, δεδομένου ότι, αν και αυτές χρησιμοποιήθηκαν εκτενώς στο παρελθόν, έχει πλέον σημειωθεί εκτεταμένη χρήση προσεγγίσεων Μηχανικής Μάθησης και Βαθιάς Μάθησης. Συνεπώς, έχουμε εστιάσει την προσοχή μας σε αλγορίθμους Μηχανικής Μάθησης και Βαθιάς Μάθησης, συμπεριλαμβανομένων επιβλεπόμενων αλγορίθμων (Supervised) (το ευρέως χρησιμοποιούμενο SVM και μια σειρά νευρωνικών δικτύων) και προσεγγίσεων μη επιβλεπόμενων (Unsupervised) για την ανίχνευση ανωμαλιών, όπως για παράδειγμα ο αλγόριθμος Isolation Forest.

Από την άλλη πλευρά, όσον αφορά την επαναλαμβανόμενη συμφόρηση ή συμφόρηση κατ' εξακολούθηση (recurring congestion), είναι γνωστό ότι αποτελεί κοινό πρόβλημα στον τομέα των μεταφορών, ιδίως σε αστικές περιοχές με μεγάλο κυκλοφοριακό όγκο. Αυτός ο τύπος συμφόρησης συνήθως προκύπτει λόγω συνηθισμένων προτύπων ζήτησης κυκλοφορίας, όπως οι πρωινές και βραδινές ώρες αιχμής. Σε αντίθεση με τη μη επαναλαμβανόμενη συμφόρηση, η οποία προκαλείται από απρόβλεπτα γεγονότα όπως ατυχήματα ή καιρικές διαταραχές, η επαναλαμβανόμενη συμφόρηση εμφανίζεται τακτικά και προβλέψιμα. Η παρουσία επαναλαμβανόμενης συμφόρησης όχι μόνο επηρεάζει την αποδοτικότητα του δικτύου μεταφορών, αλλά οδηγεί επίσης σε αυξημένη κατανάλωση καυσίμων, υψηλότερες εκπομπές ρύπων και μεγαλύτερους χρόνους ταξιδιού για τους μετακινούμενους. Οι τεχνικές περιγραφικής ανάλυσης, όπως η ανάλυση χρονοσειρών, ανάλυση μέσω οπτικοποίησεων και η συσταδοποίηση (clustering), παρέχουν μια θεμελιώδη κατανόηση των προτύπων συμφόρησης. Οι μέθοδοι προβλεπτικής αναλυτικής, συμπεριλαμβανομένης της ανάλυσης παλινδρόμησης, των αλγορίθμων μηχανικής μάθησης και της πρόβλεψης χρονοσειρών, επιτρέπουν την ακριβή πρόβλεψη των μελλοντικών συνθηκών κυκλοφορίας. Μαζί, αυτές οι τεχνικές προσφέρουν ένα ισχυρό πλαίσιο για την κατανόηση και τη διαχείριση της επαναλαμβανόμενης συμφόρησης, ανοίγοντας το δρόμο για πιο αποδοτικά συστήματα μεταφορών.

Η παρούσα διατριβή, εν συνεχεία, παρουσιάζει τη μεθοδολογία που προτείνει για την ανίχνευση περιστατικών η οποία στηρίζεται στη χρήση της Αυτόματης Μηχανικής Μάθησης (AutoML). Ο πρωταρχικός στόχος του AutoML είναι να μειώσει τις χειροκίνητες διαδικασίες που συνεπάγεται η χρήση τεχνολογιών μηχανικής μάθησης, επιταχύνοντας έτσι την ανάπτυξή τους. Κατά συνέπεια, διάφορα συστήματα έχουν προσπαθήσει να ελαχιστοποιήσουν την εργασία που απαιτείται για την εκτέλεση ορισμένων βημάτων της ροής εργασίας ανάπτυξης συστημάτων μηχανικής μάθησης.  Η παρούσα εργασία περιλαμβάνει την ανάπτυξη μιας μεθοδολογίας για την αυτόματη ανίχνευση περιστατικών με στόχο τον έγκαιρο εντοπισμό μη προγραμματισμένων μη επαναλαμβανόμενων περιστατικών και, συνεπώς, τη δημιουργία ενός ασφαλέστερου και πιο αξιόπιστου συστήματος διαχείρισης ευφυών μεταφορών. Το διάγραμμα ροής της μεθοδολογίας, που απεικονίζεται στο Σχήμα 0-6, απεικονίζει τη γενική ροή εργασίας της εν λόγω προσέγγισης. Αρχικά, η διαδικασία ξεκινά με την εισαγωγή δεδομένων, ακολουθούμενη από ένα στάδιο προεπεξεργασίας των δεδομένων, ώστε το σύνολο δεδομένων να καταστεί κατάλληλο για την ανάπτυξη του μοντέλου. Στη συνέχεια, ένα εργαλείο/ framework AutoML, το TPOT, χρησιμοποιείται ως βάση της προσέγγισής μας για την ανάπτυ, προσαρμογή και τελικά την επιλογή των πλέον κατάλληλων μοντέλων.

Η χρήση του AutoML παρέχει ένα λειτουργικό πλεονέκτημα όσον αφορά τη λεπτομερή ρύθμιση των παραμέτρων. Το TPOT, με τη βελτιστοποίηση που βασίζεται στον γενετικό προγραμματισμό, μπορεί να εξερευνήσει επαναληπτικά τον χώρο των παραμέτρων για τη λεπτομερή ρύθμιση του μοντέλου καθώς εισέρχονται νέα δεδομένα ή καθώς αλλάζουν οι συνθήκες κυκλοφορίας, μια διαδικασία που είναι πιο αποδοτική ως προς τους πόρους και ενδεχομένως πιο αποτελεσματική από τις προσπάθειες χειροκίνητης ρύθμισης. Επομένως, ενώ το μοντέλο που επιλέγεται μέσω του TPOT μπορεί να είναι σταθερό κατά τη διάρκεια μιας συγκεκριμένης περιόδου, η μεθοδολογία μας έχει σχεδιαστεί για να διευκολύνει την εξέλιξη του μοντέλου, επιτρέποντας συνεχείς βελτιώσεις και την ενσωμάτωση νέων δεδομένων, γεγονός που αποτελεί σημαντικό πλεονέκτημα σε σχέση με μια στατική αλγοριθμική προσέγγιση.

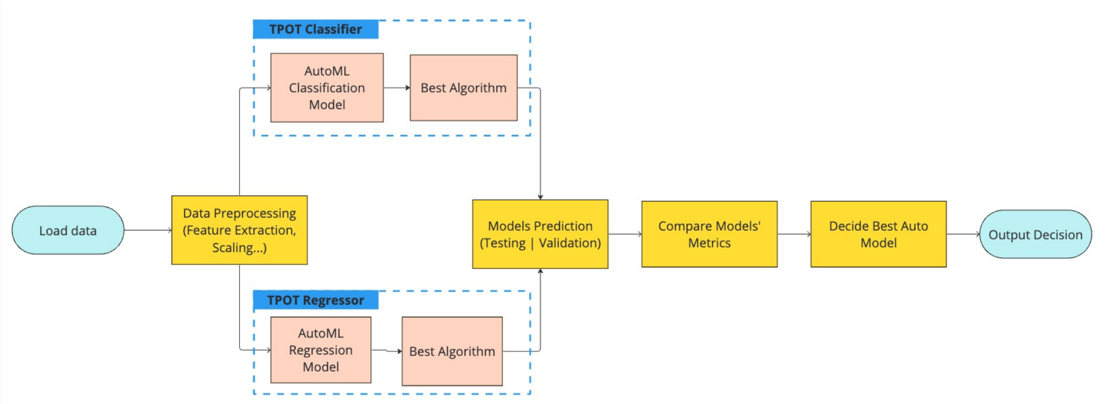



**Figure 0-6: Γενικό διάγραμμα ροής της προτεινόμενης μεθοδολογίας με AutoML.**

Είναι σημαντικό να τονιστεί ότι το στάδιο της προεπεξεργασίας των δεδομένων - το οποίο περιλαμβάνει την εξαγωγή χαρακτηριστικών, τη δειγματοληψία δεδομένων και το normalization- πραγματοποιείται πριν από την εκπαίδευση των μοντέλων. Αυτή η προεπεξεργασία είναι αναπόσπαστο μέρος της διαδικασίας σχεδιασμού και των δύο μοντέλων (classification και regression) - ωστόσο, έχει ληφθεί η απόφαση η προεπεξεργασία να εκτελείται ανεξάρτητα για να διασφαλιστεί η ομοιομορφία μεταξύ των μοντέλων και, τελικά, να ενισχυθεί η αποτελεσματικότητά τους. Ένα λεπτομερές διάγραμμα, όπως φαίνεται στο Σχήμα 0-7, παρέχει μια εις βάθος άποψη της φάσης μοντελοποίησης, απεικονίζοντας τα περίπλοκα βήματα που εμπλέκονται στην εκπαίδευση τόσο του μοντέλου ταξινόμησης όσο και του μοντέλου παλινδρόμησης, αναδεικνύοντας έτσι τη διπλή προσέγγιση της αντιμετώπισης της αυτόματης ανίχνευσης περιστατικών.

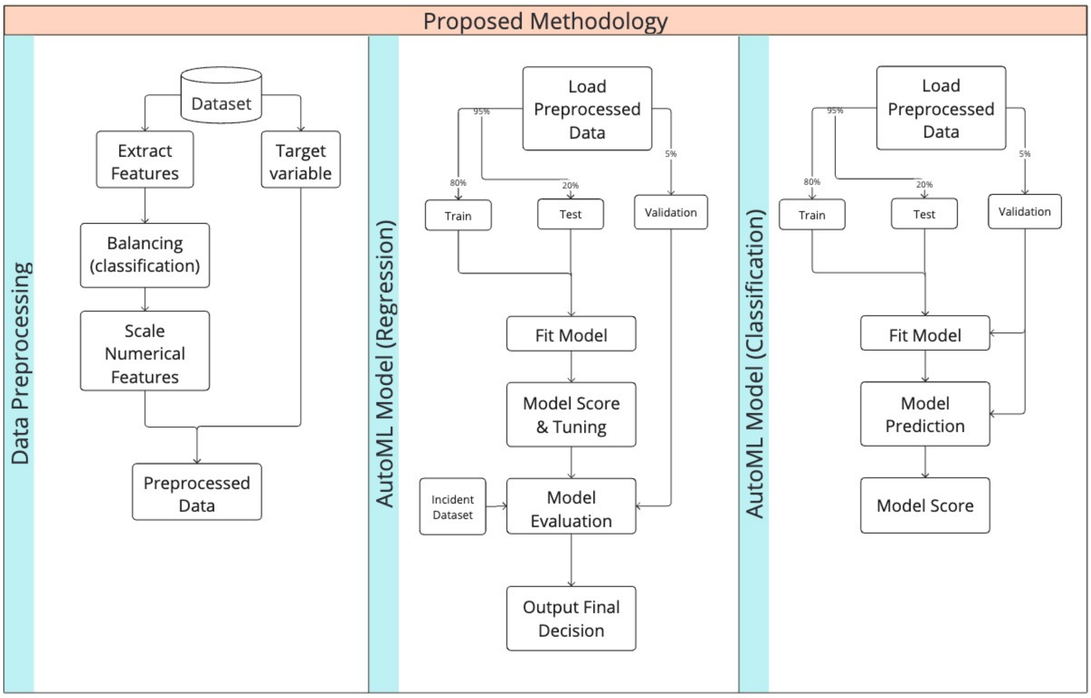



**Figure 0-7: Λεπτομερής επισκόπηση της φάσης μοντελοποίησης.**

Κατά την παρούσα διατριβή, επίσης, διερευνάται η ενσωμάτωση μηχανισμών human-in-the-loop σε συστήματα ανίχνευσης τροχαίων περιστατικών με βάση την τεχνητή νοημοσύνη. Η ανθρώπινη παρέμβαση είναι απαραίτητη όχι μόνο για τη διασφάλιση της ακριβούς απόδοσης των μοντέλων τεχνητής νοημοσύνης, αλλά και για την προώθηση της διαφάνειας, της εμπιστοσύνης και της σιγουριάς μεταξύ των ανθρώπινων χειριστών. Η επεξηγηματικότητα διαδραματίζει κρίσιμο ρόλο σε αυτή τη διαδικασία, βοηθώντας τους χειριστές να κατανοήσουν γιατί τα μοντέλα ΤΝ και τα μοντέλα που βασίζονται σε δεδομένα παράγουν συγκεκριμένες προβλέψεις για τα κυκλοφοριακά περιστατικά. Μέσω σαφών επεξηγήσεων των χαρακτηριστικών, των παραγόντων και της λογικής πίσω από αυτές τις προβλέψεις, οι χειριστές του συστήματος είναι καλύτερα εξοπλισμένοι για να παρέχουν τεκμηριωμένη ανατροφοδότηση. Αυτή η ανατροφοδότηση τους επιτρέπει να αποδέχονται, να απορρίπτουν ή να επεξεργάζονται τις λεπτομέρειες των περιστατικών που επισημαίνονται από το σύστημα, γεγονός που βελτιώνει έτσι τη διαδικασία πρόβλεψης. Ως αποτέλεσμα, αυτή η δυναμική αλληλεπίδραση μεταξύ της ανθρώπινης επίβλεψης και των συστημάτων ΤΝ ενισχύει τόσο την ακρίβεια όσο και την προσαρμοστικότητα της ανίχνευσης κυκλοφοριακών περιστατικών, διασφαλίζοντας ότι το σύστημα βελτιώνεται με βάση την πραγματική ανθρώπινη εμπειρογνωμοσύνη.

Η ανάπτυξη της ΤΝ σε κρίσιμες εφαρμογές όπως η ανίχνευση κυκλοφοριακών περιστατικών απαιτεί προσεκτική ισορροπία μεταξύ αυτοματοποίησης και ανθρώπινης εποπτείας. Ενώ τα μοντέλα τεχνητής νοημοσύνης προσφέρουν πρωτοφανείς δυνατότητες επεξεργασίας και ανάλυσης μεγάλων συνόλων δεδομένων για τον εντοπισμό περιστατικών, η πολυπλοκότητα και η αδιαφάνεια αυτών των μοντέλων συχνά δημιουργούν προκλήσεις όσον αφορά την εμπιστοσύνη και την αξιοπιστία. Οι μεθοδολογίες Human-in-the-loop (HITL) παρέχουν μια πρακτική λύση σε αυτές τις προκλήσεις, ενσωματώνοντας την ανθρώπινη ανατροφοδότηση και διασφαλίζοντας την ακρίβεια των προβλέψεων της ΤΝ. Επιπλέον, η ενσωμάτωση χαρακτηριστικών επεξηγηματικότητας στο σύστημα συμβάλλει στην ενίσχυση της εμπιστοσύνης στη διαδικασία ανίχνευσης των συστημάτων ΤΝ.

Όταν ένα περιστατικό εντοπίζεται από το σύστημα, ο χειριστής καλείται να αναγνωρίσει το περιστατικό, επιβεβαιώνοντας την εμφάνισή του. Αυτός ο βρόχος ανατροφοδότησης (feedback loop) διασφαλίζει ότι ελαχιστοποιούνται τα ψευδώς θετικά (false positive) αποτελέσματα και ότι οι προβλέψεις του συστήματος συμφωνούν με τα πραγματικά δεδομένα. Επιπλέον, εάν συμβεί ένα περιστατικό και το σύστημα δεν το αναφέρει, οι χειριστές μπορούν να εισάγουν χειροκίνητα αυτές τις πληροφορίες, διασφαλίζοντας ότι δεν παραβλέπονται κρίσιμα περιστατικά. Αυτή η αμφίδρομη αλληλεπίδραση όχι μόνο βελτιώνει την ακρίβεια του συστήματος, αλλά παρέχει επίσης πολύτιμα δεδομένα για την επανεκπαίδευση και τη βελτίωση των μοντέλων τεχνητής νοημοσύνης με την πάροδο του χρόνου.

Ένα άλλο κίνητρο είναι η ανάγκη για επεξηγηματικότητα και αξιοπιστία στα συστήματα τεχνητής νοημοσύνης, ειδικά σε περιπτώσεις όπου οι αποφάσεις που λαμβάνονται από τα μοντέλα ΤΝ μπορεί να έχουν σημαντικές επιπτώσεις στη δημόσια ασφάλεια και τη διαχείριση των πόλεων. Η ενσωμάτωση τεχνικών όπως LIME (Local Interpretable Model-agnostic Explanations) και SHAP (SHapley Additive exPlanations) μας επιτρέπει να παρέχουμε διαφανείς και κατανοητές προβλέψεις σε συστήματα που χρησιμοποιούν ΤΝ. Αυτές οι τεχνικές βοηθούν στην αποκάλυψη της λογικής πίσω από τις αποφάσεις της ΤΝ, καθιστώντας ευκολότερο για τους ανθρώπινους χειριστές να εμπιστεύονται και να βασίζονται στο σύστημα. Εξασφαλίζοντας ότι οι προβλέψεις της ΤΝ δεν είναι μόνο ακριβείς αλλά και επεξηγήσιμες, μπορούμε να προωθήσουμε μεγαλύτερη εμπιστοσύνη και υιοθέτηση αυτών των τεχνολογιών σε πραγματικές συνθήκες.

Το προτεινόμενο πλαίσιο για τη βελτίωση των συστημάτων ανίχνευσης περιστατικών ενσωματώνει τόσο τις μεθοδολογίες Human-in-the-Loop (HITL) όσο και χαρακτηριστικά επεξηγηματικότητας, συνδυάζοντας τα πλεονεκτήματα της τεχνητής νοημοσύνης με την ανθρώπινη εμπειρογνωμοσύνη, ώστε να διασφαλίζεται ακριβής και αξιόπιστη απόδοση. Αξιοποιώντας το HITL, το σύστημα επιτρέπει τη συνεχή ανθρώπινη παρέμβαση και εποπτεία, επιτρέποντας στους εμπειρογνώμονες να επικυρώνουν και να βελτιώνουν τα αποτελέσματα που παράγονται από την ΤΝ. Αυτή η υβριδική προσέγγιση διασφαλίζει ότι το σύστημα μπορεί να μαθαίνει δυναμικά από την ανθρώπινη ανατροφοδότηση, παρέχοντας παράλληλα ερμηνεία των προβλέψεων μέσω χαρακτηριστικών επεξηγηματικότητας. Αυτοί οι μηχανισμοί επεξηγηματικότητας είναι κρίσιμοι για την ενίσχυση της εμπιστοσύνης στα συστήματα που βασίζονται στην ΤΝ, καθώς επιτρέπουν στους ανθρώπινους χειριστές να κατανοήσουν το σκεπτικό πίσω από τις αποφάσεις, να διαγνώσουν πιθανά σφάλματα και να κάνουν προσαρμογές για τη βελτίωση της ακρίβειας του συστήματος. Τελικά, αυτό το πλαίσιο αποσκοπεί στην ενίσχυση της αποδοτικότητας ανίχνευσης μη επαναλαμβανόμενων περιστατικών, όπως οι διακοπές κυκλοφορίας ή τα ατυχήματα, διατηρώντας παράλληλα υψηλά επίπεδα διαφάνειας, εμπιστοσύνης και απόδοσης σε συστήματα μεταφορών του πραγματικού κόσμου.

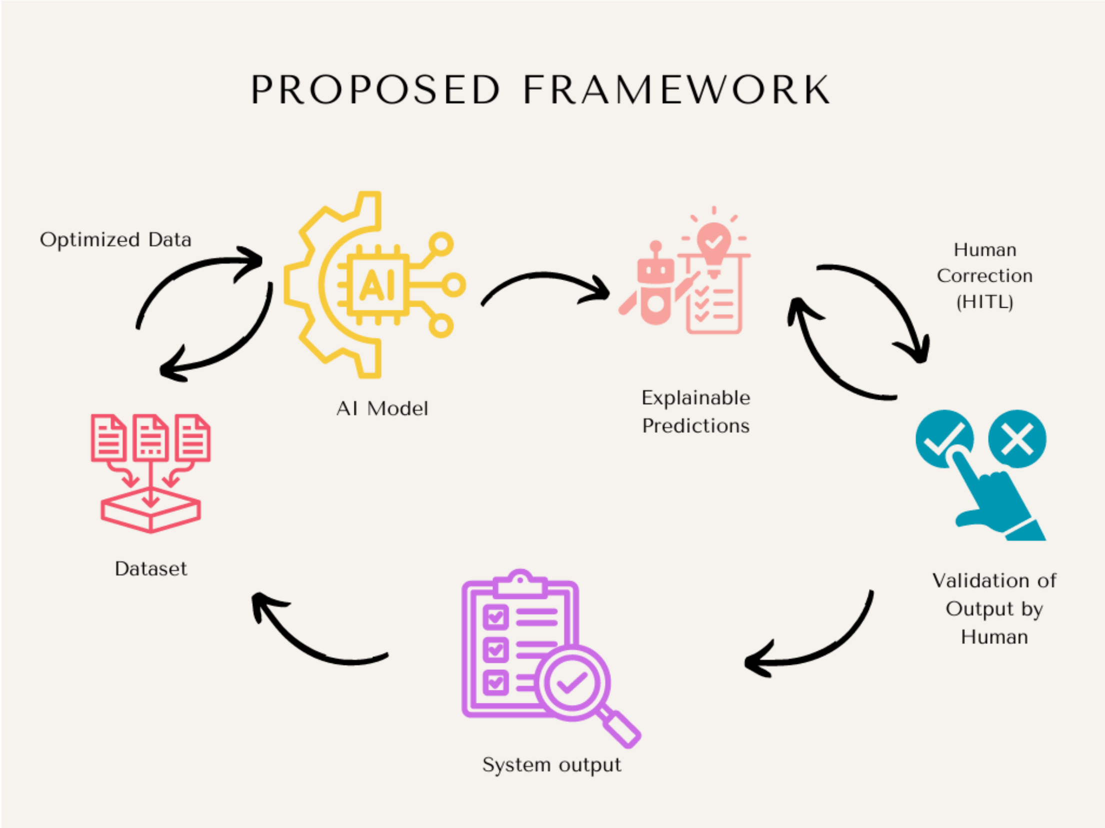



**Figure 0-8: Το προτεινόμενο πλαίσιο με τεχνικές Επεξηγησιμότητας και Human-in-the-Loop.**

Υπάρχουν πολλά οφέλη του προτεινόμενου πλαισίου (Figure 0-8), όπως για παράδειγμα ενισχυμένη ακρίβεια και αξιοπιστία, αυξημένη διαφάνεια και εμπιστοσύνη, συνεχής μάθηση και προσαρμογή κατά τη διάρκεια προληπτικής ή real-time διαχείρισης της κυκλοφορίας. Συνοπτικά, το προτεινόμενο πλαίσιο για την ενσωμάτωση προσεγγίσεων Human-in-the-Loop και χαρακτηριστικών επεξηγηματικότητας σε συστήματα ανίχνευσης περιστατικών προσφέρει μια καλή λύση για την αποτελεσματική διαχείριση της κυκλοφορίας. Συνδυάζοντας τα πλεονεκτήματα της τεχνητής νοημοσύνης με την ανθρώπινη εμπειρογνωμοσύνη και τη διαφανή λήψη αποφάσεων, το σύστημα εξασφαλίζει ακριβή και αξιόπιστη ανίχνευση περιστατικών, συμβάλλοντας τελικά σε ασφαλέστερη διαχείριση της κυκλοφορίας.

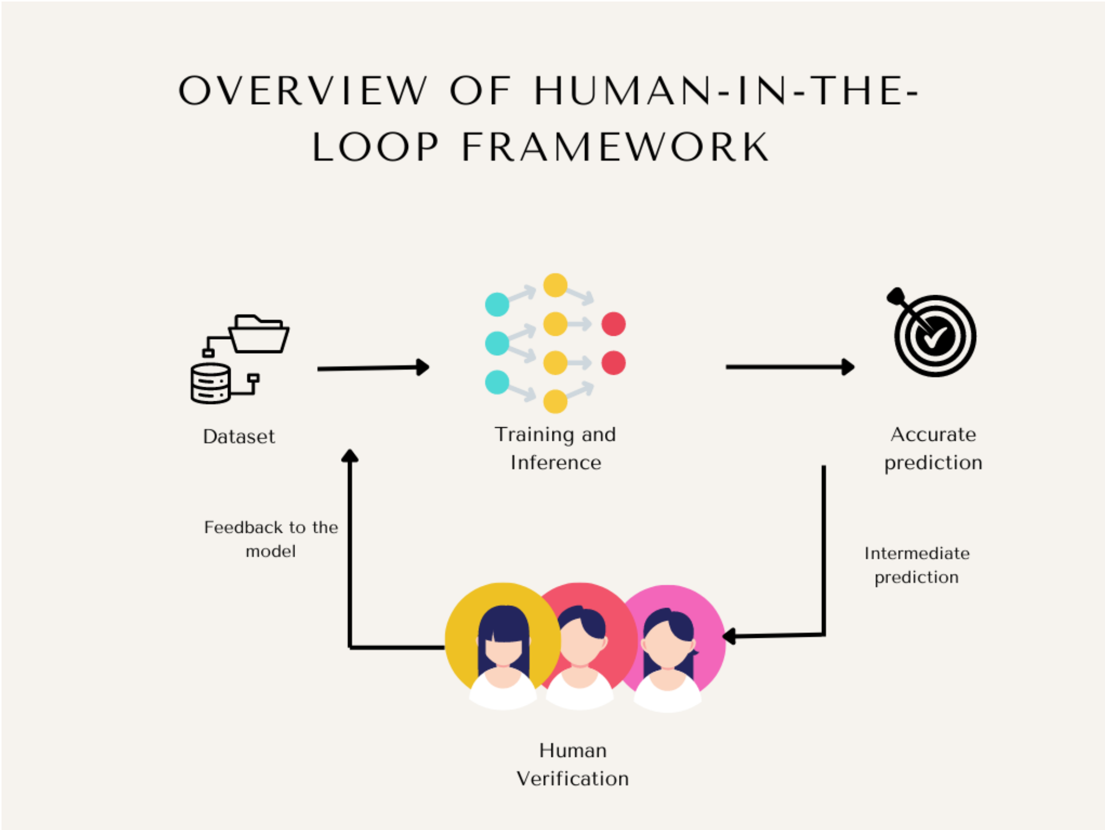



**Figure 0-9: Το προτεινόμενο πλαίσιο εστιάζοντας στο Human -in-the-Loop approach**

Το Σχήμα 0-9 απεικονίζει το πλαίσιο Human-in-the-Loop (HITL), μια μεθοδολογία που χρησιμοποιείται στην έρευνά μας για τη βελτίωση των μοντέλων μηχανικής μάθησης (ML) για την πρόβλεψη περιστατικών σε ευφυή συστήματα μεταφορών που χρησιμοποιούν δεδομένα αισθητήρων βρόχου (inductive loop detectors - ILD) . Η διαδικασία ξεκινά με τη συλλογή δεδομένων από αισθητήρες βρόχου ενσωματωμένους σε οδούς, τα οποία στη συνέχεια χρησιμοποιούνται για την εκπαίδευση μοντέλων ML με στόχο την πρόβλεψη περιστατικών. Αντί να βασίζεται αποκλειστικά σε αυτοματοποιημένες προβλέψεις, η προσέγγιση HITL εισάγει ένα ενδιάμεσο βήμα όπου ανθρώπινοι εμπειρογνώμονες εξετάζουν και επικυρώνουν αυτές τις προβλέψεις. Η ανατροφοδότηση από αυτούς τους εμπειρογνώμονες στη συνέχεια τροφοδοτείται εκ νέου στο μοντέλο, βελτιώνοντας περαιτέρω την ακρίβειά του και επιτρέποντάς του να προσαρμόζεται στις πολυπλοκότητες των σεναρίων του πραγματικού κόσμου. Αυτή η επαναληπτική διαδικασία διασφαλίζει ότι τα μοντέλα ML όχι μόνο βελτιώνονται προοδευτικά, αλλά και συγκλίνουν όλο και περισσότερο με τον τρόπο που οι επαγγελματίες των μεταφορών ορίζουν την έννοια του «περιστατικού», οδηγώντας τελικά σε ακριβέστερες και πιο αξιόπιστες προβλέψεις περιστατικών.

Επιπροσθέτως, το σύστημά μας έχει δυνατότητα για επανεκπαίδευση των μοντέλων μηχανικής μάθησης σε τακτά χρονικά διαστήματα, προκειμένου να διατηρείται η αποτελεσματικότητά του σε σχέση με τις μετρικές που έχουν τεθεί, και καθώς νέα δεδομένα γίνονται διαθέσιμα. Μετά την ολοκλήρωση των διαδικασιών επανεκπαίδευσης, διαπιστώθηκε πως οι μετρικές απόδοσης του μοντέλου παρουσιάζουν αξιοσημείωτη βελτίωση μετά από κάθε κύκλο επανεκπαίδευσης. Η βελτίωση αυτή σημειώθηκε ιδιαίτερα στη μετρική recall, η οποία μετρά την ικανότητα του μοντέλου να αναγνωρίζει σωστά τα πραγματικά περιστατικά, συμπεριλαμβανομένων εκείνων που είχαν προηγουμένως ταξινομηθεί εσφαλμένα. Επιπλέον, συγκρίνοντας συστηματικά τα αποτελέσματα της εβδομαδιαίας και της δεκαπενθήμερης επανεκπαίδευσης, εντοπίσαμε την πιο αποτελεσματική στρατηγική επανεκπαίδευσης. Δεν επιλέξαμε την καθημερινή επανεκπαίδευση, καθώς ενώ μπορεί να οδηγήσει σε ταχεία αύξηση των επιδόσεων, θα μπορούσε επίσης να ενέχει τον κίνδυνο υπερπροσαρμογής και ταυτόχρονα απαιτεί σημαντικούς υπολογιστικούς πόρους. Η εβδομαδιαία επανεκπαίδευση παρέχει μια ισορροπημένη προσέγγιση, προσφέροντας συνεχείς βελτιώσεις χωρίς υπερβολικές υπολογιστικές απαιτήσεις. Η δεκαπενθήμερη επανεκπαίδευση, αν και δυνητικά πιο αποδοτική ως προς τους πόρους, έχει αποδειχθεί ότι καθυστερεί την ικανότητα του μοντέλου να ενσωματώνει άμεσα νέα πρότυπα. Η συγκριτική ανάλυση των συχνοτήτων επανεκπαίδευσης αποκάλυψε ότι η εβδομαδιαία επανεκπαίδευση παρείχε τη βέλτιστη ισορροπία μεταξύ ανταπόκρισης και σταθερότητας. Το μοντέλο ήταν σε θέση να προσαρμοστεί αποτελεσματικά σε νέα πρότυπα χωρίς τον κίνδυνο υπερβολικής προσαρμογής ή υπερβολικών υπολογιστικών απαιτήσεων.

Ακολούθως, αναπτύσσεται πληροφοριακό σύστημα, το AutoEventX, το οποίο δίνει τη δυνατότητα εφαρμογής της προτεινόμενης μεθόδου σε διαφορετικά αστικά περιβάλλοντα. Συγκεκριμένα, η μέθοδος εφαρμόζεται σε δύο διαφορετικές πόλεις, την Αθήνα και την Αμβέρσα, για να αξιολογηθεί και να συγκριθεί σε διαφορετικά αστικά περιβάλλοντα και πλαίσια. Η εννοιολογική αρχιτεκτονική (conceptual architecture) απεικονίζεται στο Σχήμα 0-10.

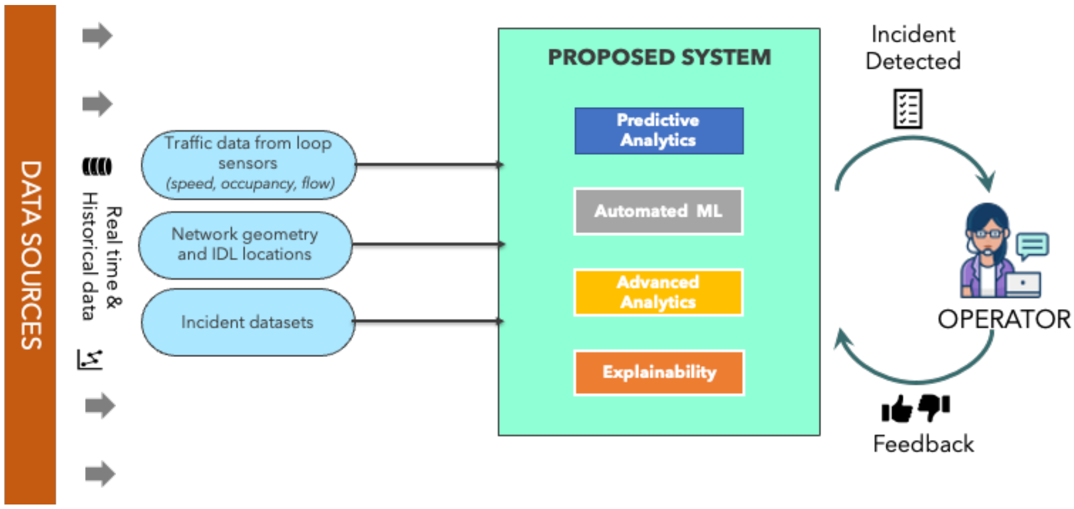



**Figure 0-10: Εννοιολογική αρχιτεκτονική του προτεινόμενου συστήματος.**

Το πληροφοριακό σύστημα αναπτύχθηκε μέσω της ενσωμάτωσης και ενοποίησης διαφόρων εργαλείων που αφορούν τις διάφορες φάσεις του προτεινόμενου πλαισίου. Το σύστημα είναι σε θέση να ενσωματώνει δεδομένα που παρέχονται από διαφορετικές πηγές, να αξιολογεί την ποιότητα των εισερχόμενων δεδομένων, μέσω ειδικών τεχνικών, ιδίως όσον αφορά τις μετρήσεις που λαμβάνονται από ανιχνευτές βρόχων, να υποστηρίζει την αποτελεσματική επεξεργασία δεδομένων σε πραγματικό χρόνο, να παρέχει στους ενδιαφερόμενους φορείς προβλέψεις όσον αφορά προγραμματισμένα και μη προγραμματισμένα περιστατικά και να αποκαλύπτει τη λογική πίσω από αυτά τα αποτελέσματα, λαμβάνοντας υπόψη την ανατροφοδότηση των εμπειρογνωμόνων χειριστών. Προκειμένου να αναπτυχθεί το σύστημα αυτό, χρησιμοποιήθηκε κατά βάση η γλώσσα Python καθώς και σχετικές βιβλιοθήκες, οι οποίες παρουσιάζονται πιο αναλυτικά στην παρούσα διδακτορική διατριβή.

Η τεχνική αρχιτεκτονική του συστήματος μαζί με τα επιμέρους επίπεδα (Αποθήκευσης, Λογικής και Ανθρώπινης Παρέμβασης) και τις μεταξύ τους συσχετίσεις παρουσιάζονται στο Σχήμα 0-11.

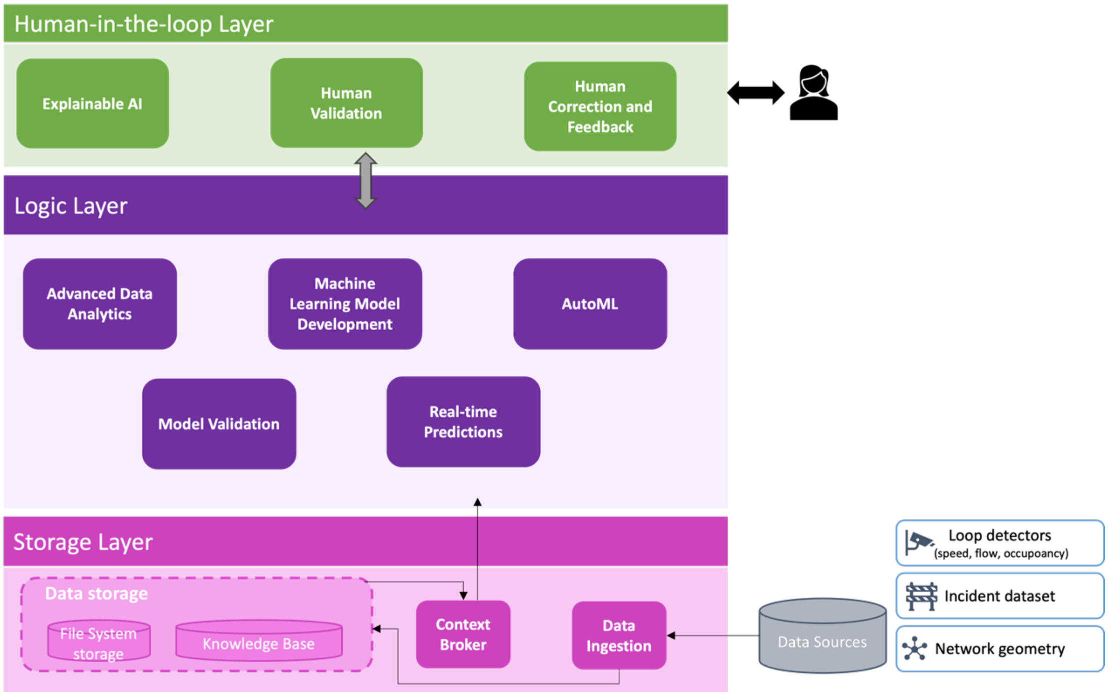



**Figure 0-11: Τεχνική αρχιτεκτονική του αναπτυχθέντος συστήματος AutoEventX.**

Το Επίπεδο Αποθήκευσης (Storage Layer) παρέχει τη δυνατότητα αποθήκευσης τόσο για στατικά δεδομένα όσο και για δεδομένα σχεδόν πραγματικού χρόνου με διαφορετικές μορφές και τύπους πρόσβασης, κι αποτελείται από αποθήκευση σε σύστημα αρχείων που διατηρεί τα δεδομένα σε μορφές JSON, JSON-LD και Parquet. Το τελευταίο έχει επιλεγεί για τη διατήρηση των αρχικών δεδομένων για περαιτέρω τροφοδοσία του συστήματος και αποκατάσταση της βάσης δεδομένων, εάν απαιτείται.

Το Επίπεδο Λογικής (Logic Layer) αποτελεί το κεντρικό υπολογιστικό επίπεδο για ανάλυση δεδομένων, εκπαίδευση μοντέλων και πρόβλεψη. Ενσωματώνει παραδοσιακούς και αυτόματους αλγόριθμους μηχανικής μάθησης για εκτέλεση προηγμένων αναλύσεων για προγραμματισμένα και μη προγραμματισμένα περιστατικά.

- **Προηγμένη Ανάλυση Δεδομένων**: Χρησιμοποιεί χρονοσειρές, χωροχρονική ανάλυση και ανάλυση συσχετίσεων, με εργαλεία όπως ARIMA και τη βιβλιοθήκη GeoPandas.

- **Ανάπτυξη Μοντέλων Μηχανικής Μάθησης**: Περιλαμβάνει προεπεξεργασία, καθαρισμό, εξαγωγή χαρακτηριστικών, επιλογή αλγορίθμου, εκπαίδευση και προσαρμογή παραμέτρων μοντέλων μηχανικής μάθησης.

- **Επικύρωση Μοντέλου**: Χρησιμοποιεί ευρέως χρησιμοποιούμενες μετρικές (π.χ. precision, recall, F1-score) και τεχνικές διασταυρούμενης επικύρωσης για εξασφάλιση αξιοπιστίας.

- **Αυτοματοποιημένη Μηχανική Μάθηση (AutoML)**: Χρησιμοποιεί βιβλιοθήκες AutoML για την αυτοματοποίηση της διαδικασίας, ενισχύοντας την αποδοτικότητα και την επεκτασιμότητα (scalability) των μοντέλων.

- **Προβλέψεις σε Πραγματικό Χρόνο**: Παρέχει προβλέψεις σε πραγματικό χρόνο για τυχόν περιστατικά, προγραμματισμένα ή μη.

Το Επίπεδο Human-in-the-Loop ενσωματώνει την ανθρώπινη εμπειρογνωμοσύνη στο σύστημα για επιβεβαίωση της προβλέψεων των μοντέλων και απόκτησης εμπιστοσύνης από τους χρήστες στην ΤΝ και τις αποφάσεις που λαμβάνει το σύστημα.

- **Επεξηγηματικότητα**: Προσφέρει πληροφορίες σχετικά με τη διαδικασία λήψης αποφάσεων του μοντέλου, επιτρέποντας στους ενδιαφερόμενους να κατανοούν τις προβλέψεις του συστήματος.

- **Επικύρωση, Διόρθωση και Ανατροφοδότηση από Ανθρώπους**: Οι χειριστές του συστήματος διαχείρισης κυκλοφορίας αναθεωρούν και διορθώνουν τις προβλέψεις του μοντέλου, δημιουργώντας έναν κύκλο ανατροφοδότησης για συνεχή βελτίωση της απόδοσης του μοντέλου.

- **Ενσωμάτωση με Συστήματα Διαχείρισης Κυκλοφορίας**: Τα επικυρωμένα αποτελέσματα μπορούν να ενσωματωθούν σε πραγματικά συστήματα διαχείρισης κυκλοφορίας, παρέχοντας ειδοποιήσεις σε πραγματικό χρόνο και διασφαλίζοντας αποτελεσματική αντίδραση σε περίπτωση ατυχημάτων.

Η προγραμματιστική γλώσσα Python, λόγω της ευελιξίας της, επιλέχθηκε για την υλοποίηση του πληροφοριακού συστήματος. Βιβλιοθήκες όπως Pandas διευκολύνουν τη διαχείριση δεδομένων, η βιβλιοθήκη Scikit-learn υποστηρίζει την ανάπτυξη μοντέλων, οι Keras και TensorFlow χρησιμοποιούνται για την ανάπτυξη μοντέλων βαθιάς μάθησης, ενώ οι Seaborn και Matplotlib για την οπτικοποίηση δεδομένων, προβλέψεων, αναλύσεων και αποτελεσμάτων. Επιπλέον, τεχνολογίες που χρησιμοποιήθηκαν περιλαμβάνουν τις βιβλιοθήκες SHAP και LIME για επεξήγηση των μοντέλων, καθώς και Flask και Docker για τη δημιουργία ενός φορητού και επεκτάσιμου συστήματος. Το Flask, ένα Python framework, επιτρέπει την ανάπτυξη API και τη διαχείριση αιτημάτων. Το Docker δημιουργώντας containers για το deployment της εφαρμογής, εξασφαλίζει συνέπεια μεταξύ περιβαλλόντων, απομόνωση για ασφάλεια και φορητότητα μεταξύ πλατφορμών.

Το σύστημά μας έχει αναπτυχθεί έχοντας 2 διαφορετικούς τρόπους λειτουργίας: την offline και την online. Στο Σχήμα 0-12 παρουσιάζεται η τεχνική αρχιτεκτονική της λειτουργίας εκτός σύνδεσης του συστήματος που αναπτύξαμε.

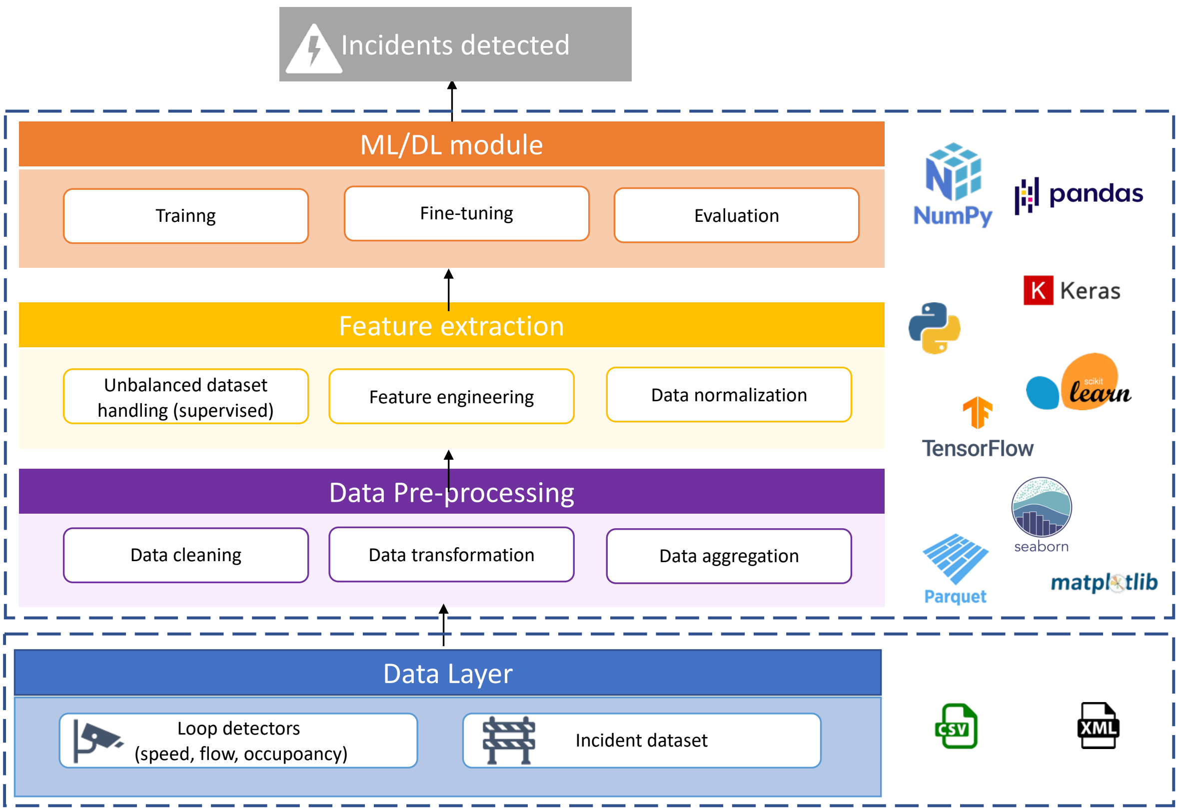


**Figure 0-12: Τεχνική αρχιτεκτονική του offline τρόπου λειτουργίας του συστήματος.**

Το επίπεδο δεδομένων περιέχει τις μετρήσεις του ανιχνευτή βρόχων (ιστορικού και πραγματικού χρόνου) για την ταχύτητα, την πληρότητα και τη ροή, εκτός από τα αντίστοιχα σύνολα δεδομένων περιστατικών και τις αντίστοιχες πληροφορίες για το συγκοινωνιακό δίκτυο κάθε περίπτωσης. Στο επίπεδο ML/DL, έχουμε υλοποιήσει μια σειρά αλγορίθμων μηχανικής μάθησης, βαθιάς μάθησης και αλγορίθμων AutoML για την αυτόματη ανίχνευση περιστατικών. Αυτές περιλαμβάνουν τόσο εποπτευόμενες όσο και μη εποπτευόμενες προσεγγίσεις.

Αφού εκτελέσουμε την εκπαίδευση του μοντέλου τεχνητής νοημοσύνης κατά τη λειτουργία offline, το σύστημά μας είναι σε θέση να λειτουργήσει σε πραγματικό χρόνο για την εμφάνιση πιθανών περιστατικών (alerts). Στο Σχήμα 0-13 παρουσιάζεται η ροή της διαδικασίας της λειτουργίας πραγματικού χρόνου του συστήματός μας.  Μόλις γίνουν διαθέσιμα νέα δεδομένα, το σύστημα τα καταγράφει. Επομένως, οι αντίστοιχες πληροφορίες συλλέγονται, αποθηκεύονται τοπικά και στη συνέχεια ομαδοποιούνται σε κατάλληλα χρονικά διαστήματα για να τροφοδοτηθούν στο στάδιο προεπεξεργασίας και καθαρισμού των δεδομένων. Οι συγκεκριμένες διαδικασίες προεπεξεργασίας παραμένουν συνεπείς με εκείνες που περιγράφονται στον offline τρόπο λειτουργίας, διατηρώντας την ομοιομορφία στην προσέγγιση της προετοιμασίας και του καθαρισμού των δεδομένων. Στη συνέχεια, τα δεδομένα μετατρέπονται στην απαιτούμενη μορφή για να τροφοδοτηθούν στο στάδιο της πρόβλεψης του μοντέλου. Εάν η καταχώρηση περιέχει ανωμαλίες (αντιπροσωπεύεται ως «1»), τότε ζητείται ανατροφοδότηση από τους χειριστές για να επιβεβαιωθεί το εντοπισμένο περιστατικό. Αυτή η έννοια της ανατροφοδότησης από ανθρώπινο χειριστή είναι ζωτικής σημασίας, δεδομένου ότι βοηθά στη δημιουργία ενός βελτιωμένου συνόλου δεδομένων περιστατικών και έτσι εξασφαλίζει ότι η απόδοση του συστήματος μπορεί να βελτιωθεί με την πάροδο του χρόνου, δεδομένου ότι επανεκπαιδεύεται σε αυτό το διαρκώς εξελισσόμενο σύνολο δεδομένων. Αξίζει να αναφερθεί ότι οι ενδιαφερόμενοι φορείς μπορούν να βελτιώσουν την ποιότητα και την ακρίβεια των αναφερόμενων περιστατικών, δημιουργώντας χειροκίνητες καταχωρίσεις περιστατικών που εντοπίζουν οι ίδιοι. Τέλος, στην περίπτωση που το σύστημα έχει διαπιστώσει μια ανωμαλία στα δεδομένα και την χαρακτηρίζει ως περιστατικό, τότε παράγει ως εκροή (output) και αποθηκεύει στο Data Layer μια οντότητα τύπου «***Περιστατικό***» με τα χαρακτηριστικά της τοποθεσίας και του χρόνου του περιστατικού.

Προκειμένου να βελτιωθούν οι δυνατότητες ανίχνευσης του συστήματος με την πάροδο του χρόνου, ο βρόχος ανατροφοδότησης που έχουμε εφαρμόσει για τη σύγκριση των προβλέψεων του μοντέλου με τα πραγματικά αποτελέσματα είναι το κλειδί για συνεχή βελτίωση. Η ανίχνευση τυχόν αποκλίσεων αξιοποιείται για τη βελτιστοποίηση του μοντέλου.

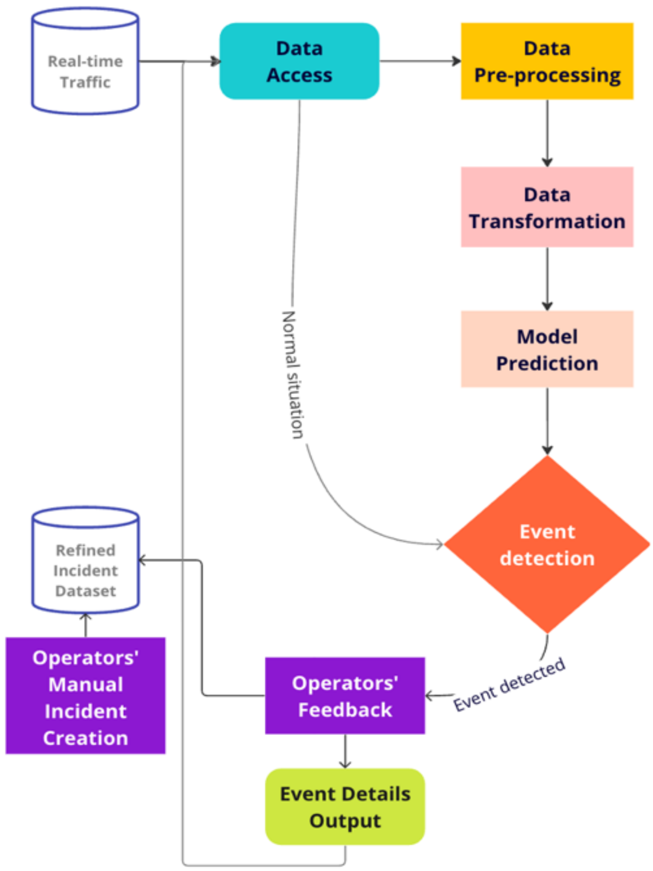



**Figure 0-13: Online τρόπος λειτουργίας του συστήματός μας.**

Έπειτα, αξιολογείται το προτεινόμενο σύστημα ανίχνευσης περιστατικών σε δύο πραγματικά σενάρια χρήσης. Η αξιολόγηση περιλαμβάνει πειράματα για τη συλλογή μετρήσεων επιδόσεων και τη διεξαγωγή συγκριτικών αναλύσεων μεταξύ των αλγορίθμων μηχανικής και βαθιάς μάθησης, τεχνικών AutoML και βασικών μοντέλων σε δύο μεγάλες πόλεις, την Αθήνα στην Ελλάδα, και την Αμβέρσα στο Βέλγιο.

Στην Αθήνα, ο αυτοκινητόδρομος της Αττικής οδού που συνδέει το αεροδρόμιο με το κέντρο της πόλης παρέχει ιδιαίτερες κυκλοφοριακές προκλήσεις, ενώ στην Αμβέρσα, μια σημαντική διαδρομή που συνδέεται με το λιμάνι και τους αυτοκινητόδρομους αντιπροσωπεύει ένα διαφορετικό αστικό περιβάλλον. Η ανάπτυξη των μοντέλων και η εφαρμογή του προτεινόμενου πλαισίου, μεθοδολογίας και αναπτυχθέντος συστήματος μας και στις δύο πόλεις μας επέτρεψε να δοκιμάσουμε την προσαρμοστικότητα σε διαφορετικά αστικά τοπία, δείχνοντας τις δυνατότητες του προτεινόμενου συστήματός μας για χρήση και σε ευρύτερες εφαρμογές διαχείρισης της κυκλοφορίας.

Ακολουθεί screenshot στο Σχήμα 0-14 που απεικονίζει ένα μη προγραμματισμένο περιστατικό κυκλοφορίας που εντοπίστηκε από το σύστημα AutoEventX όπως αυτό φαίνεται από dashboard που αναπτύχθηκε στο πλαίσιο του ερευνητικού έργου FRONTIER.

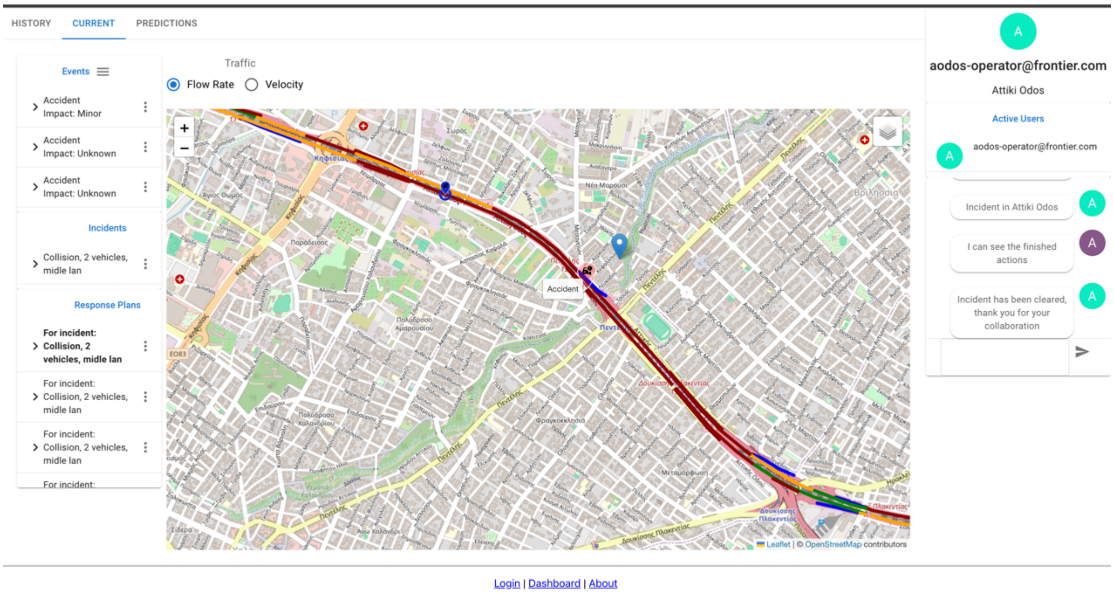



**Figure 0-14: Στιγμιότυπο από μη προγραμματισμένο περιστατικό στο case study στην Αθήνα.**

Κατά την αξιολόγηση λήφθηκαν υπόψη οι περιορισμοί των διαθέσιμων δεδομένων, ιδίως η έλλειψη επισημασμένων περιστατικών, η οποία περιόριζε την ορατότητα και επηρέαζε το ποσοστό ψευδώς θετικών αποτελεσμάτων. Η ανίχνευση στηρίχθηκε σε ακριβή διαστήματα 5 λεπτών, γεγονός που επηρέασε τις μετρικές απόδοσης, καθώς το σύστημα έπρεπε να ανιχνεύσει περιστατικά με ακριβή timestamps.

Τα αποτελέσματα έδειξαν διακυμάνσεις μεταξύ των αλγορίθμων, με το Support Vector Machine (SVM) να επιτυγχάνει την υψηλότερη ακρίβεια και ανάκληση και στα δύο σύνολα δεδομένων, γεγονός που συνάδει με τη βιβλιογραφία που υποστηρίζει την υψηλή αποτελεσματικότητα απόδοσης του SVM με επισημειωμένα δεδομένα. Ωστόσο, οι περιορισμοί περιλαμβάνουν πιθανή υπερπροσαρμογή και προκλήσεις στο χειρισμό δειγμάτων που δεν υπήρχαν στο training dataset. Τα BCNN και Wavelet Neural Networks επέδειξαν υψηλή ανάκληση αλλά χαμηλότερη ακρίβεια, επηρεάζοντας το F1-score. Οι μετασχηματισμοί wavelet είχαν ελαφρώς καλύτερες επιδόσεις από το BCNN, σύμφωνα με τα βιβλιογραφικά ευρήματα για δεδομένα χρονοσειρών. Ο Autoencoder έχει περιθώρια βελτίωσης, όμως η απλή αρχιτεκτονική που επιλέχθηκε πιθανώς περιόρισε την απόδοση. Ο αλγόριθμος Isolation Forest πέτυχε καλή ανάκληση αλλά χαμηλή ακρίβεια, αποδίδοντας πολυάριθμα ψευδώς θετικά αποτελέσματα, τα οποία ήταν δύσκολο να αξιολογηθούν λόγω πιθανών τυφλών σημείων του δικτύου για τα οποία δεν υπήρχαν καθόλου δεδομένα ή σε περίπτωση που υπήρχαν, αυτά ήταν ελλιπή. Ο αλγόριθμος Bidirectional LSTM παρουσίασε υψηλή ακρίβεια και ικανοποιητική ανάκληση. Η βέλτιστη απόδοση επιτεύχθηκε μέσω μιας βαθιάς bidirectional αρχιτεκτονικής, αναλύοντας αποτελεσματικά τις εξαρτήσεις προς τα εμπρός και προς τα πίσω για την πρόβλεψη της ροής κυκλοφορίας. Το Random Forest είναι σε θέση να αποδώσει εξαιρετικά καλά τόσο στις μετρήσεις ακρίβειας όσο και στις μετρήσεις ανάκλησης, αναδεικνύοντας την ικανότητά του να ταξινομεί με ακρίβεια τα πραγματικά περιστατικά, ελαχιστοποιώντας παράλληλα τα ψευδώς θετικά και αρνητικά. Το Νευρωνικό Δίκτυο Γραφημάτων (Graph Neural Network – GNN) είναι σε θέση να καταγράψει πολύ ικανοποιητικές τιμές ακρίβειας και αρκετά καλές τιμές ανάκλησης. Από πρακτική άποψη, αυτό σημαίνει ότι υπάρχουν λίγα false alarms όπου το σύστημα προβλέπει ένα περιστατικό που δεν έχει συμβεί στην πραγματικότητα, ενώ είναι ικανό να αναγνωρίσει ένα μεγάλο ποσοστό των πραγματικών περιστατικών. Αυτό σημαίνει ότι το σύστημα είναι αξιόπιστο με την έννοια ότι δεν χάνει πολλά περιστατικά. Τέλος, το baseline μοντέλο της Aimsun, αν και αδύναμο σε ακριβείς μετρήσεις ανά πέντε λεπτά, πέτυχε 73% ανάκληση σε ένα περιθώριο 15 λεπτών γύρω από τα γεγονότα στην Αθήνα, το οποίο ήταν αποδεκτό για μη επαναλαμβανόμενα περιστατικά. Επιπλέον, αναφερόμενοι σύντομα στα αποτελέσματα της ανάλυσης χρησιμοποιώντας AutoML, στην Αθήνα, η προσέγγιση υπερείχε των βασικών μεθόδων όσον αφορά όλες τις καθιερωμένες μετρήσεις. Αντίθετα, στην Αμβέρσα, ο αλγόριθμος SVM ήταν ανώτερος όσον αφορά το F1-score και την ανάκληση, αλλά ταυτόχρονα η προσέγγιση χρησιμοποιώντας AutoML τον ξεπέρασε όσον αφορά τη μετρική της ακρίβειας. Αυτές οι διαφοροποιήσεις υπογραμμίζουν την ανάγκη για προσαρμοσμένες αλγοριθμικές στρατηγικές και την εξέταση των ιδιαιτεροτήτων των δεδομένων κατά την ανίχνευση περιστατικών σε διαφορετικά αστικά περιβάλλοντα.

Τέλος, είναι σημαντικό να αναγνωριστεί ότι, ενώ το AutoML στοχεύει στην απλοποίηση και τη βελτιστοποίηση της διαδικασίας επιλογής και εκπαίδευσης μοντέλων, δεν αναιρεί την αξία της κατανόησης της απόδοσης συγκεκριμένων τεχνικών ML. Η σύγκρισή μας επιδιώκει να αναδείξει τον τρόπο με τον οποίο η προσέγγισή μας που βασίζεται σε τεχνικές AutoML αποδίδει έναντι των χειροκίνητα ρυθμισμένων και επιλεγμένων μοντέλων στον τομέα της ανίχνευσης περιστατικών χρησιμοποιώντας δεδομένα ανιχνευτών βρόχων, δίνοντας έμφαση στην αποτελεσματικότητα, την προσαρμοστικότητα και την απόδοση σε σενάρια πραγματικού κόσμου. Συνεπώς, καταλήγουμε πως το AutoML βοηθά στην επιλογή μοντέλων, ειδικά κατά τα πρώτα στάδια της μοντελοποίησης, και γενικώς μπορεί να δώσει πολύ καλές και συγκρίσιμες επιδόσεις, από την άλλη μεριά απαιτεί πολύ χρόνο και υπολογιστικούς πόρους.

Από τα παραπάνω είναι σαφές πως η ανάλυσή μας αναδεικνύει τις προκλήσεις στην ελαχιστοποίηση των ψευδώς θετικών αποτελεσμάτων που οφείλονται σε τυφλά σημεία των συγκοινωνιακών δικτύων. Κάθε πόλη και σενάριο χρήσης έχει ξεχωριστές προκλήσεις, υπογραμμίζοντας την ανάγκη για προσαρμοσμένες αλγοριθμικές στρατηγικές. Να σημειωθεί επίσης πως οι επιλεγμένες μετρικές, όπως η ακρίβεια, η ανάκληση και το F1-score, παρείχαν πολύτιμες πληροφορίες, ωστόσο πρόσθετες μετρικές -όπως ο μέσος χρόνος ανίχνευσης περιστατικών, και η ταχύτητα απόκρισης- θα ήταν καλό να αναπτυχθούν στο μέλλον για ένα ολοκληρωμένο operational context, που εκτείνεται πέρα από την παρούσα διατριβή.

Στο τελευταίο κεφάλαιο, η διατριβή αποτυπώνει τις σημαντικές συνεισφορές στον τομέα της ανίχνευσης περιστατικών κυκλοφορίας με βάση την τεχνητή νοημοσύνη. Συνδυάζοντας τις τεχνικές AutoML, HITL και την επεξηγηματικότητα, το σύστημα αναδεικνύει τη σημαντική πρόοδο που προτείνει σε σχέση με τις παραδοσιακές μεθόδους ανίχνευσης περιστατικών. Ωστόσο, οφείλουμε να αναγνωρίσουμε κάποιους περιορισμούς, ιδίως όσον αφορά την ποιότητα των δεδομένων και τις προκλήσεις του χειρισμού ελλιπών ή θορυβωδών συνόλων δεδομένων.

Αυτοί οι περιορισμοί υποδεικνύουν τομείς για μελλοντική έρευνα. Η μελλοντική έρευνα θα μπορούσε να επικεντρωθεί στη βελτίωση της ανίχνευσης κυκλοφοριακών περιστατικών με τεχνητή νοημοσύνη μέσω ενσωμάτωσης δεδομένων από πολλαπλές πηγές, συνδυάζοντας δεδομένα από κάμερες κλειστού κυκλώματος (CCTV), δεδομένα δημόσιων συγκοινωνιών, δεδομένα καιρού και πληροφορίες από δίκτυα κοινωνικής δικτύωσης για ένα πιο ολοκληρωμένο σύστημα. Επιπλέον, οι εξελίξεις στη μηχανική μάθηση, βαθιά μάθηση και ενισχυτική μάθηση μπορούν πιθανώς να βελτιστοποιήσουν την προσαρμοστικότητα των μοντέλων σε διάφορα αστικά περιβάλλοντα. Η συνεργασία ανθρώπου-τεχνητής νοημοσύνης θα μπορούσε να βελτιωθεί μέσω διαδραστικών διεπαφών, εργαλείων επαυξημένης/εικονικής πραγματικότητας (AR/VR) και ανατροφοδότησης σε πραγματικό χρόνο, διασφαλίζοντας την ομαλή ενσωμάτωση μεταξύ αυτοματοποιημένων προβλέψεων και λήψης αποφάσεων από τους ανθρώπους. Παράλληλα, η χρήση πιο προηγμένων τεχνικών επεξηγησιμότητας είναι πιθανό να προσφέρει μεγαλύτερη διαφάνεια στα μοντέλα AI, διευκολύνοντας τους διαχειριστές κυκλοφορίας στην κατανόηση των προβλέψεων και στη λήψη τεκμηριωμένων αποφάσεων. Επιπρόσθετα, η ενσωμάτωση προγνωστικών αναλύσεων θα βοηθήσει στη βελτιστοποίηση της κυκλοφορίας, προτείνοντας μέτρα για τη μείωση της συμφόρησης σε πραγματικό χρόνο. Επιπλέον, τα συστήματα ανίχνευσης περιστατικών θα μπορούσαν να συνδεθούν με ευρύτερες υποδομές έξυπνων πόλεων, ενισχύοντας τη συνδεσιμότητα της τεχνητής νοημοσύνης με τις υπηρεσίες έκτακτης ανάγκης και τη λήψη μέτρων για την βελτίωση της κυκλοφοριακής συμφόρησης.

Συνοψίζοντας, η έρευνα που παρουσιάζεται στην παρούσα διατριβή θέτει τις βάσεις για μια νέα γενιά συστημάτων διαχείρισης οδικής κυκλοφορίας, που είναι πιο αποτελεσματικά, διαφανή και αξιόπιστα. Συνδυάζοντας τα πλεονεκτήματα της τεχνητής νοημοσύνης και της αναλυτικής δεδομένων με την ανθρώπινη εμπειρία και παρέμβαση, το προτεινόμενο σύστημα στοχεύει να βελτιώσει σημαντικά την ασφάλεια και την αποτελεσματικότητα της κυκλοφορίας σε αστικά περιβάλλοντα.

# List of Acronyms

| | |
|---|---|
| **ACF** | Autocorrelation Function |
| **ACT** | Adaboost-Cart |
| **ADF** | Augmented Dickey-Fuller |
| **AI** | Artificial Intelligence |
| **AID** | Automatic Incident Detection |
| **AIDA** | Automatic Incident Detection Algorithms |
| **AL** | Active Learning |
| **ANN** | Artificial Neural Network |
| **API** | Application Programming Interface |
| **AR** | Augmented Reality |
| **ARIMA** | Autoregressive Integrated Moving Average |
| **AutoML** | Automated Machine Learning |
| **AV** | Autonomous Vehicles |
| **BCNN** | Bayesian Convolutional Neural Network |
| **BiLSTM** | Bidirectional Long Short Term Memory |
| **BJSON** | Binary-JSON |
| **CB** | Context Broker |
| **CCAM** | Connected, Cooperative, And Automated Mobility |
| **CCTV** | Closed-Circuit Television |
| **CNN** | Convolutional Neural Network |
| **CSV** | Comma-Separated Values |
| **DBN** | Dynamic Bayesian Network |
| **DL** | Deep Learning |
| **DNN** | Deep Neural Networks |
| **DR-Score** | Discrimination And Reconstruction Anomaly Score |
| **DS** | Data Storage |
| **DT** | Decision Tree |
| **DTGN** | Differential Time-Varying Graph Neural Network |
| **EDA** | Exploratory Data Analysis |
| **EL** | Ensemble Learning |
| **ENISA** | European Union Agency For Cybersecurity |
| **ERD** | Entity-Relationship Diagram |
| **ERT** | Emergency Roadside Telephones |
| **ETL** | Extract Transform Load |
| **ETSI** | European Telecommunications Standardizations Institute§ |
| **FRONTIER** | Next-Generation Traffic Management For Cavs Integration And Multimodal Optimization |
| **GA** | Genetic Algorithm |
| **GAN** | Generative Adversarial Network |
| **GCN** | Graph Convolutional Network |
| **GNN** | Graph Neural Network |
| **GTFS** | General Transit Feed Specification |

| | |
|---|---|
| **HAIM-DRL** | Human As AI Mentor-Based Deep Reinforcement Learning |
| **HITL** | Human-In-The-Loop |
| **HITLML** | Human-In-The-Loop Machine Learning |
| **HPO** | Hyper Parameter Optimization |
| **ICT** | Information And Communication Technology |
| **IDA** | Incident Detection Algorithm |
| **IF** | Isolation Forest |
| **ILD** | Inductive Loop Detector |
| **IML** | Interactive Machine Learning |
| **ITS** | Intelligent Transport System |
| **JSON** | Javascript Object Notation |
| **kNN** | K-Nearest Neighbors |
| **LD** | Linked Data |
| **LIME** | Local Interpretable Model-Agnostic Explanations |
| **LoS** | Level Of Service |
| **LPU** | Local Processing Unit |
| **LSTM** | Long Short Term Memory |
| **ML** | Machine Learning |
| **MSP** | Model Selection Problem |
| **MT** | Machine Teaching |
| **MTME** | Multimodal Traffic Management Ecosystem |
| **MTTD** | Mean Time to Detect |
| **NaN** | Not A Number |
| **NB** | Naïve Bayes |
| **NGSI** | Next Generation Service Interface |
| **NGSI-LD** | Next Generation Service Interface - Linked Data |
| **NN** | Neural Network |
| **OCB** | Orion Context Broker |
| **PACF** | Partial Autocorrelation Function |
| **PPCA** | Probabilistic Principal Component Analysis |
| **Pro-Graph** | Propagation Graph |
| **RBF** | Radial Basis Function |
| **RDF** | Resource Description Framework |
| **RFID** | Radio-Frequency Identification |
| **RNN** | Recurrent Neural Network |
| **RQ** | Research Question |
| **SGD** | Stochastic Gradient Descent |
| **SHAP** | Shapley Additive Explanations |
| **SMOTE** | Synthetic Minority Oversampling Technique |
| **SND** | Standard Normal Deviate |
| **STC** | Spatiotemporal Congestion |
| **STCCP** | Spatiotemporal Congestion Co-Location Pattern |
| **STL** | Seasonal-Trend Decomposition Using Loess |
| **STOTree** | Spatiotemporal Outlier Tree |
| **SVM** | Support Vector Machine |
| **TF** | Traffic Forecasting |

| | |
|---|---|
| **TMC** | Traffic Management Center |
| **TPOT** | Tree-Based Pipeline Optimization Tool |
| **UML** | Unified Modelling Language |
| **VR** | Virtual Reality |
| **WA** | Wavelet Analysis |
| **WNN** | Wavelet Neural Network |
| **XAI** | Explainable Artificial Intelligence |
| **XML** | Extensible Markup Language |

# Glossary

| | |
|---|---|
| **Artificial Intelligence** | Τεχνητή Νοημοσύνη |
| **Autoencoder** | Αυτοκωδικοποιητής |
| **Autoencoder** | Αυτοκωδικοποιητής |
| **Automated Machine Learning** | Αυτοματοποιημένη Μηχανική Μάθηση |
| **Bayesian Neural Network** | Νευρωνικό Δίκτυο Bayes |
| **Bias** | Μεροληψία, Προκατάληψη |
| **Bidirectional LSTM** | Αμφίδρομο LSTM |
| **Bottleneck** | Σημείο Συμφόρησης |
| **Classification** | Ταξινόμηση |
| **Conceptual Architecture** | Εννοιολογική Αρχιτεκτονική |
| **Congestion** | Συμφόρηση |
| **Cross-validation** | Διασταυρούμενη Επικύρωση |
| **Data Analytics** | Αναλυτική Δεδομένων |
| **Data Preprocessing** | Προεπεξεργασία Δεδομένων |
| **Data-driven Approach** | Προσέγγιση Βασισμένη Σε Δεδομένα |
| **Decision Tree** | Δέντρο Απόφασης |
| **Deep Learning** | Βαθιά Μάθηση |
| **Ensemble Learning** | Συνολική Μάθηση |
| **Explainability** | Επεξηγησιμότητα |
| **F1-score** | Μέτρο F1 |
| **False Positive** | Ψευδώς Θετικό |
| **Feature Engineering** | Σχεδιασμός Χαρακτηριστικών |
| **Feature Extraction** | Εξαγωγή Χαρακτηριστικών |
| **Feedback Loop** | Βρόχος Ανατροφοδότησης |
| **Flow Rate** | Ρυθμός Ροής |
| **Gradient Boosting** | Ενίσχυση Βαθμίδας |
| **Graph Neural Network (GNN)** | Νευρωνικό Δίκτυο Γράφων |
| **Human-in-the-Loop** | Ανθρώπινη Παρέμβαση |
| **Hyperparameter** | Υπερπαράμετρος |
| **Hyperparameter Tuning** | Προσαρμογή Υπερπαραμέτρων |
| **Incident Classification** | Κατηγοριοποίηση Περιστατικών |
| **Incident Management** | Διαχείριση Περιστατικών |
| **Inductive Loop Detector (ILD)** | Ανιχνευτές Βρόχου |
| **Information System** | Πληροφοριακό Σύστημα |
| **Intelligent Decision Support Systems** | Ευφυή Συστήματα Υποστήριξης Απόφασης |
| **Intelligent Transport Systems (ITS)** | Ευφυή Συστήματα Μεταφορών |
| **LIME (Local Interpretable Model-Agnostic Explanations)** | Τοπικά Ερμηνεύσιμες Εξηγήσεις Μοντέλων |
| **Long Short-Term Memory (LSTM)** | Μνήμη Μακράς και Βραχείας Διάρκειας |
| **Machine Learning** | Μηχανική Μάθηση |

| | |
|---|---|
| **Model Explainability** | Επεξηγησιμότητα Μοντέλου |
| **Model Optimization** | Βελτιστοποίηση Μοντέλου |
| **Model Training** | Εκπαίδευση Μοντέλου |
| **Model Validation** | Επικύρωση Μοντέλου |
| **Multimodal Data** | Πολυτροπικά Δεδομένα |
| **Neural Network** | Νευρωνικό Δίκτυο |
| **Non-recurring Congestion** | Μη Επαναλαμβανόμενη Συμφόρηση |
| **Occupancy** | Πληρότητα |
| **Online Learning** | Επιγραμμική Μάθηση |
| **Overfitting** | Υπερπροσαρμογή |
| **Precision** | Ακρίβεια |
| **Predictive Analytics** | Προβλεπτική Αναλυτική |
| **Queue** | Ουρά Αναμονής |
| **Random Forest** | Τυχαίο Δάσος |
| **Real-time** | Σε Πραγματικό Χρόνο |
| **Regression** | Παλινδρόμηση |
| **System Deployment** | Ανάπτυξη και Εγκατάσταση Συστήματος |
| **Recall** | Ανάκληση |
| **Recurring Congestion** | Επαναλαμβανόμενη Συμφόρηση Ή Συμφόρηση Κατ’ Εξακολούθηση |
| **Retraining** | Επανεκπαίδευση |
| **Scalability** | Επεκτασιμότητα |
| **Semi-supervised Learning** | Ημι-Εποπτευόμενη Μάθηση |
| **SHAP (Shapley Additive Explanations)** | Επεξηγήσεις Shapley |
| **Smart City** | Έξυπνη Πόλη |
| **Spatio-temporal** | Χωροχρονικός |
| **Spatiotemporal Analysis** | Χωροχρονική Ανάλυση |
| **Supervised Learning** | Εποπτευόμενη Μάθηση |
| **Time Series Analysis** | Ανάλυση Χρονοσειρών |
| **Timeseries** | Χρονοσειρές |
| **Traffic Data Visualization** | Οπτικοποίηση Δεδομένων Κυκλοφορίας |
| **Traffic Demand** | Κυκλοφοριακή Ζήτηση |
| **Traffic Flow** | Ροή Κυκλοφορίας |
| **Traffic Incident Detection** | Ανίχνευση Κυκλοφοριακών Περιστατικών |
| **Traffic Sensors** | Αισθητήρες Κυκλοφορίας |
| **Unsupervised Learning** | Μη Εποπτευόμενη Μάθηση |
| **Urban Traffic Management** | Διαχείριση Αστικής Κυκλοφορίας |
| **Wavelet Transformation** | Μετασχηματισμός Κυματιδίων (Wavelet) |

# 1 Introduction

## 1.1 Motivation

In an era characterized by rapid technological advancements and the increasing interconnectivity of systems, the ability to detect and respond to emerging situations—whether planned or unplanned—has become crucial across various sectors. From disaster management to cybersecurity, and from public health to urban planning, the need for timely and accurate incident detection is more pressing than ever. Especially in transportation systems, the need for efficient and timely incident identification and mitigation is critical for the safety, resilience and effective management of the transport system. Traditional methods of incident detection, which often rely on predefined rules and manual monitoring struggle with scalability, adaptability, and accuracy, often failing in dynamic urban environments.

To address these challenges, the integration of Machine Learning (ML) and automated Machine Learning (AutoML) techniques offers a transformative approach. These technologies enable the automatic identification of patterns and anomalies within large, even vast, datasets, facilitating the detection of emerging situations with greater speed and accuracy than manual methods. However, the deployment of purely automated systems presents significant challenges, particularly concerning transparency, trust, and the alignment of machine-generated insights with human intuition and expertise. Throughout this Thesis, a few such challenges have been explored and suggested answers have been provided.

The concept of Human-in-the-Loop (HITL) in machine learning systems addresses these issues by incorporating human oversight and interaction into the automated processes. This approach not only enhances the explainability of the system but also ensures that the insights generated are interpretable and actionable by human users. The integration of HITL frameworks is particularly important in domains where the stakes are high, and decisions based on system outputs can have important consequences.

The motivation for developing an AI-driven HITL-enabled incident detection framework and information system stems from the pressing need for innovative solutions that can effectively detect and respond to emerging situations across various domains. By integrating explainability features, this approach addresses the critical issue of trust in automated systems, providing users with clear and understandable insights into the system's decision-making processes. This transparency is essential for fostering confidence in the system and ensuring that it can be reliably used in high-stakes environments, including but not limited to the intelligent transportation sector.

Moreover, the incorporation of AutoML techniques facilitates the continuous improvement and adaptability of the system. AutoML allows for the automatic optimization of ML models, ensuring that the system remains effective even as data patterns evolve over time. This capability is particularly important in dynamic environments, like those of transportation systems, where the nature of emerging situations can change rapidly.

The flexibility of a HITL-enabled, autoML-driven event detection system makes it applicable to a wide range of fields, not only in transportation systems, offering significant potential to improve outcomes in areas such as emergency response and infrastructure monitoring. The system's ability to adapt to different scenarios and provide actionable insights in real-time positions it as a valuable tool for organizations looking to enhance their situational awareness and decision-making capabilities.

In summary, the development of an incident detection system that integrates ML, autoML, explainability and HITL is driven by the need to create a robust, reliable, and user-centric tool capable of detecting and responding to emerging situations in the field of transportation. By addressing the limitations of existing incident detection systems and incorporating advanced ML techniques within a HITL framework, this approach aims to set a new standard in intelligent event detection and response. The questions addressed in this study have brought Machine Learning (ML), automated Machine Learning (autoML), and Human-in-the-Loop (HITL) approaches into play. This doctoral dissertation attempts to propose a comprehensive framework and provide answers and suggestions that pave the way for the effective detection and response to emerging situations and ultimately contributing to a more resilient and responsive future of transportation in urban environments.

## *1.2 Contribution*

The present Thesis is positioned within the context of automatic incident detection, intersecting with the research pathways of Artificial Intelligence (AI), automated Machine Learning (autoML), and Human-in-the-Loop (HITL) methodologies. These approaches aim to enhance decision-making processes for system operators and provide explainable insights to improve traffic management.

The contributions of this present Thesis can be detailed as follows:

- **Model Integration and Optimization**: The present Thesis integrates diverse data sources (traffic measurements from loop detectors, incidents datasets and network topology information) into ML and DL models. This methodology enhances the accuracy and reliability of automatic incident detection on urban highways, ensuring that operators can make informed decisions.
- **AutoML Methodologies Integration:** The Thesis leverages autoML techniques to automatically optimize ML models. AutoML automates the process of selecting, configuring, and tuning machine learning algorithms. This includes hyperparameter tuning, model selection, and feature engineering, which ensures that the most effective models are used for incident detection without extensive manual intervention. This continuous optimization allows the system to adapt to changing traffic patterns and emerging trends, maintaining high performance and accuracy over time.
- **End-to-end System Development and Real-World Case Study Deployment:** A novel information system has been developed and deployed in real-world case

studies to demonstrate the practical application of the proposed models and methodologies. Two detailed use-case scenarios are provided to demonstrate the effectiveness of the proposed approach in aiding operators for effective decision-making and incident management on urban highways.

- **Explainability Features Inclusion:** The proposed system includes explainability features, ensuring that operators can understand the reasoning and rationale behind the system's outcomes and predictions. This transparency is crucial for building trust and facilitating informed decision-making.
- **Human-in-the-Loop (HITL) Integration:** By embedding HITL components, the system ensures continuous human oversight and interaction. This allows for real-time adjustments and improvements based on operator input, enhancing the overall effectiveness and adaptability of the system in urban environments.

In summary, this Thesis makes significant contributions by developing a robust framework for detecting events on urban highways using advanced AI and autoML techniques, combined with HITL methodologies. It provides practical tools and methodologies for system operators to enhance their decision-making processes and delivers explainable, actionable insights, thus contributing to more efficient and trustworthy urban transportation systems.

## *1.3 Relation to scientific publications*

During the research evolved within this present Ph.D. Thesis, several scientific papers have been published in scientific conferences and international journals leading to the progress of the herein demonstrated study. In the below paragraphs, a summary of each publication that assisted to the documenting of the herein Chapters follows, while a thorough catalog of the publications is available in the "List of Publications" Section. Moreover, further details about how the scientific publications respond to the respective Research Questions this Thesis poses are presented in Section 3.2.

The journal paper [j2] in addition to the conference papers [c2] and [c3] provide foundational ideas on how advanced data-driven techniques can be utilized to enhance incident detection in intelligent transportation systems. These papers are integral to Chapter 2, which provides a Literature Review of the investigated issues.

The publication [c2], [c3] and [j2] set the groundwork for the proposed system developed in this Thesis. They demonstrate early work on AI-based methods for automatic incident detection, leveraging heterogeneous multimodal big data. The paper [j2] presents a stable version of the system, built and tested in experimental and real-life pilot conditions, showcasing the practical application of the AutoML-based approach for traffic incident detection. The aforementioned papers are positioned in Chapters 6, 8 and 9 where the proposed methodology, capabilities and information system is presented.

Moreover, the publications [c1], [c4] and [j3] illustrates the technical achievements in terms of the whole system and is positioned in Chapters 5, 7, 8 and 9, where the framework, machine learning and deep learning models are presented in addition to the actual developed system with the respective use cases where it has been deployed and assessed. Lastly, results and conclusions are briefly discussed in these publications but further explored and expanded in the current dissertation.

## *1.4 Relation to Research Project*

The current Ph.D. thesis has been partially funded by the European Commission Research Project with the title FRONTIER (Next generation traffic management for empowering CAVs integration, cross-stakeholders collaboration, and proactive multi-modal network optimization). This project is part of the Horizon 2020 Research and Innovation Framework Program under Grant Agreement 101006633.

- ✓ The objective of the project was to develop future integrated traffic management strategies that consider new transport modes, including automated vehicles, to minimize pollution, reduce capacity bottlenecks, lower accident rates, and decrease mobility costs for all users. The project promoted resilient multimodal autonomous mobility through stakeholder collaboration and viable business models. It implemented and tested autonomous management systems that evolve with real-time data, operator knowledge, and simulation models. FRONTIER has been validated at pilot sites in Oxfordshire (UK), Athens (GR), and Antwerp (BE), focusing on smart infrastructure, multimodal mobility, and network performance.

- ✓ A great part of the herein presented Thesis has been developed within the FRONTIER project, contributing to its goals by enhancing functionality of traffic management systems. The Thesis supplements the project by providing advanced tools and methodologies, particularly aimed at system operators and traffic managers, to improve real-time incident detection and management on urban highways. This effort aims to facilitate the practical application and success of the FRONTIER project's objectives.

## *1.5 Research Design and Structure of the Dissertation*

The research design and methodology of the present Thesis are illustrated in Figure 1-1, while an overview of each Chapter and its included components is provided in Table 1-1 below.

The dissertation corpus begins with the **Literature Review** Chapter providing comprehensive information about the domain of automated traffic incident detection. It presents the current research landscape, key technologies, methodologies, and their applications in urban settings. Next, the **Research Challenges** which the Thesis studies are presented. Next, the chapter **Framework for Real-Time Monitoring and Prediction of Traffic Incidents** introduces the conceptual framework, detailed methodology, and conceptual software architecture developed within the Thesis to tackle the challenges identified. Moreover, the chapter **AI-Driven Traffic Incident Detection for Planned and Unplanned Events** is dedicated to the examined and employed data-driven and ML/DL methods for predicting both planned and unplanned events. Subsequent chapters delve into **AutoML** techniques, discussing their application in optimizing ML models for incident detection, and **Human-in-the-Loop (HITL) / Explainable AI (XAI)**, which explain how these subcomponents enhance system transparency and reliability. The **Information System AutoEventX** chapter details the system architecture and implementation, providing an overview and detailed explanation of different layers and their functions. The **Deployment and Evaluation in Real-world Case Studies** chapter includes practical case studies from Athens, Greece, and Antwerp, Belgium, showcasing the system's real-world application and results. The dissertation concludes with **Conclusions and Future Work,** summarizing findings, discussing limitations and potential extensions, and proposing future research directions.

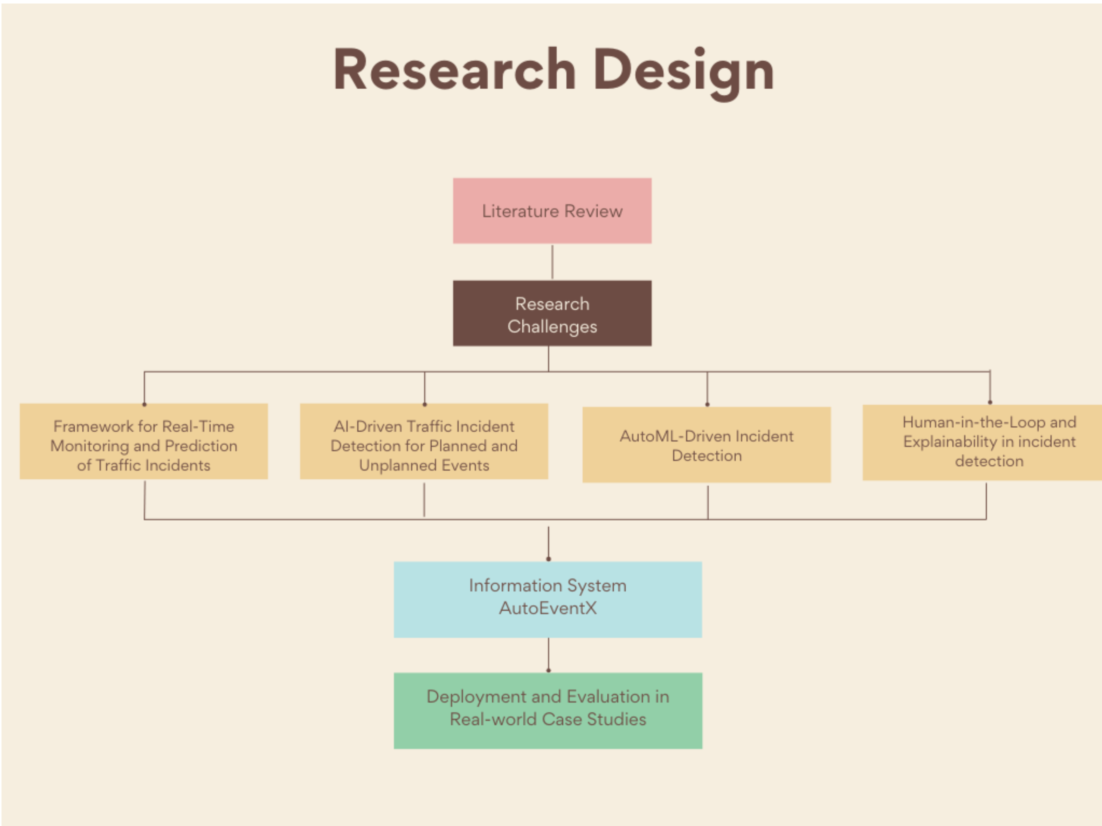



**Figure 1-1:The Research Design and Methodology**

**Table 1-1: The Components of each Research Methodology Step**

| Research Methodology Step | Components |
|---|---|
| ***Literature Review***<br>*(Chapter 2)* | • Detailed presentation of the research area and state-of-the-art material regarding: |

<table>
<tr><td></td><td><ul style="list-style-type:circle"><li>Background of Automatic Incident Detection</li><li>Review of methodologies and tools including machine learning algorithms, statistical models, and hybrid approaches used for incident detection.</li><li>Overview of key research studies and projects contributing to advancements in the field.</li><li>Challenges and research gaps identified.</li></ul></td></tr>
<tr><td>Research Challenges<br>(Chapter 3)</td><td><ul><li>Research Questions<ul style="list-style-type:circle"><li>RQ1: What are the key components and methodologies for real-time monitoring and prediction in AI-based traffic incident detection?</li><li>RQ2: How can human-centered traditional and automated AI technologies be leveraged to develop a comprehensive framework for real-time detection of traffic incidents, monitoring, and situational awareness of urban networks?</li><li>RQ3: How do AI-driven methodologies and algorithms enhance the detection of planned and unplanned traffic incidents?</li><li>RQ4: How can AutoML techniques enhance the development of AI models for traffic incident detection?</li><li>RQ5: How to ensure human in the loop and prediction is explainable and transparent in AI-based traffic incident detection systems?</li></ul></li><li>The Thesis</li></ul></td></tr>
<tr><td>Framework for Real-Time Monitoring and Prediction of Traffic Incidents<br>(Chapter 4)</td><td><ul><li>Pillars of our framework<ul style="list-style-type:circle"><li>Data Analytics</li><li>Automated Machine Learning</li><li>Explainability</li><li>Human-in-the-Loop</li></ul></li><li>Proposed methodology</li></ul></td></tr>
<tr><td>AI-Driven Traffic Incident Detection for Planned and Unplanned Events</td><td><ul><li>Introduction and Motivation</li></ul></td></tr>
</table>

| | |
|---|---|
| *(Chapter 5)* | • Data-driven Algorithms for unplanned non-recurring incident detection<br>• Advanced Analytics Methods for Recurring Congestion Identification |
| ***AutoML-Driven Incident Detection*** *(Chapter 6)* | • Introduction and Motivation<br>• State-of-the-art analysis<br>• Proposed Methodology<br>• The Implementation – Technical Details |
| ***Human-in-the-Loop and Explainability in incident detection*** *(Chapter 7)* | • Introduction and Motivation<br>• Human-in-the-Loop State-of-the-art<br>• Explainability State-of-the-art<br>• Proposed methodology |
| ***Information System AutoEventX*** *(Chapter 8)* | • System architecture and implementation<br>• Technical Architecture<br>• Modes of operation<br>• Examples of system use |
| ***Deployment and Evaluation in Real-world Case Studies*** *(Chapter 9)* | • Real-World Case Studies description<br>• Evaluation of proposed method<br>  ○ Evaluation of ML and DL models<br>  ○ Evaluation of AutoML models<br>  ○ Integration of Explainability features<br>  ○ Simulating the Retraining Process with Human Feedback<br>• Discussion and analysis of results |
| ***Conclusions and Future Work*** *(Chapter 10)* | • Conclusions of the conducted work<br>• Limitations<br>• Future work and research directions |

# 2 Literature Review

In this Chapter, a thorough literature review on traffic analysis is presented, focusing on the development and evolution of automated traffic incident detection systems. The synthesis of this emerging field and its key technologies are presented, alongside the objectives it aims to achieve and the most widely used methods and techniques. The Chapter highlights significant topics investigated in this Thesis. Additionally, a summary of related EU projects is briefly mentioned. Lastly, the Chapter discusses research gaps, limitations and future directions of automated incident detection systems in the context of advancing transportation technologies.

## *2.1 Background*

### *2.1.1 Related works on Traffic Analysis Applications*

Traffic congestion is a global issue that has been acknowledged by transportation science for over a decade. In the United States alone, drivers spend 6.9 billion hours stuck in traffic annually, wasting over 11 billion liters of fuel, as reported by INRIX (INRIX. n.d., 2024). On a per capita basis, individuals in Russia and Thailand experience even greater delays, with Brazil, South Africa, the United Kingdom, and Germany not far behind the U.S. Utlizing mobility data science and understanding the behavior of human participants across different transportation modes offers promising avenues for addressing these challenges. Two primary research areas have emerged: (1) traffic monitoring at an aggregate level to support city administration, and (2) delivering services directly to road users. (Mokbel, et al., 2024)

Research on traffic monitoring encompasses several domains, including congestion monitoring (Li, Han, Lee, & Gonzalez, 2007), road and intersection safety assessments (Maeda, Sekimoto, & Seto, 2016), traffic prediction evacuation (Li, Yu, Shahabii, & Liu, 2018), routing (Zhang, Zhang, & Guo),and public transportation schedule optimization (Richly, Teusner, Immer, Windheuser, & Wolf, 2015).

Meanwhile, services for road users focus on solutions such as traffic-aware routing to distribute load across roads (Souza, Yokoyama, Maia, Loureiro, & Villas, 2016), assisting drivers in locating nearby facilities (Kolahdouzan & Shahabi, 2004), personalized routing (Li, Gunopulos, Lu, & Guibas, 2019), eco-routing to reduce greenhouse gas emissions (Lin, Choy, Ho, Sai Ho Chung, & Lam, 2014), and multi-modal trip planning (Tomaras, Kalogeraki, Liebig, & Gunopulos, 2018). More specifically, an automated method is presented to generate and evaluate traffic incident response plans using a template library and Aimsun Next simulation. The approach optimizes responses in real-time, enhancing network performance and aiding traffic management decisions. (Almohammad & Georgakis, Automated Approach for Generating and Evaluating Traffic Incident Response Plans, 2023). Despite these advances, numerous opportunities and challenges remain in leveraging mobility data to improve traffic management. For instance, developing precise models for the dynamic scheduling of public transportation or optimizing traffic signals in context-aware ways—such as accounting for pedestrian flows near bus or train stations to reduce stop-and-go vehicle impacts—are critical areas for further exploration. A significant challenge in this domain is monitoring and reducing transportation-related emissions. Accurately quantifying emissions through data collected from in-situ sensors and remote sensing technologies, such as satellite-based earth observation, is vital for accountability and emission reduction efforts. This data can help assess the impact of e-mobility adoption, improvements in collective transportation systems, and infrastructure enhancements, ultimately supporting more sustainable and efficient traffic solutions.

Mobility data science also plays a critical role in supporting cities by enabling data-driven map construction (Ahmed, Karagiorgou, Pfoser, & Wenk, 2015) and updating existing maps to reflect blocked or newly added road segments (Chen, et al., 2016). This capability is especially vital for applications in autonomous driving (Macfarlane & Stroila, 2016). Real-time monitoring of urban mobility contributes to situational awareness—a concept originally developed in defense applications. Situational awareness involves three key components: perceiving environmental states through surrounding data, comprehending this data to understand emerging situations, and projecting future states or events through predictive analytics. Mobility data serves as a cornerstone for situational awareness in urban environments. When effectively utilized, it not only supports the development of resilient critical infrastructures but also safeguards them against threats such as forest fires, earthquakes, or terrorist attacks. Researchers have increasingly leveraged mobility data to enhance situational awareness in urban areas and specialized environments like airports (Shao, et al., 2019).

The field of mobility data analytics has grown significantly, covering diverse applications across urban mobility, maritime, aviation, and personal movement domains [ (Zhao, Tarkoma, Liu, & Vo, 2016), (Claramunt, et al., 2017), (Chung, Ma, Mark Hansen, & Choi, 2020), (Ossi, Hachem, Cagnacci, Demšar, & Damiani., 2022), (Jensen, Lu, & Yang, 2010)]. Urban mobility, as the largest area of research, addresses key challenges such as traffic anomaly detection (Pan, Zheng, Wilkie, & Shahabi., 2013), hotspot analysis (Nikitopoulos, Paraskevopoulos, Doulkeridis, Pelekis, & Theodoridis., 2018), road traffic prediction (Nag & Simon, 2018) and travel time estimation (Wang, Tang, Kuo, Kifer, & Li, 2019). Efforts in developing generic methods for mobility data analysis encompass various approaches, including trajectory clustering (Wang, Bao, Culpeppe, & Cong., 2021), trajectory similarity measures , (Toohey & Duckham., 2015) outlier detection (Han, Cheng, Ma, & Grubenmann, 2022), transportation mode classification (Biljecki, Ledoux, & Oosterom, 2013), spatiotemporal pattern detection (Sakr & Güting, 2014), and trajectory completion (Krumm., 2022. ). Despite these extensive research efforts, a unified set of tools and systems for mobility data analysis remains lacking. The landscape of scientific software for this field is notably fragmented. For instance, a review by (Joo, et al., 2020) identifies 58 R packages dedicated to movement analysis, while (Graser., 2023) examines Python libraries designed for movement data analysis and visualization.

Recent advancements in deep learning (DL) have introduced transformative approaches, such as leveraging Generative Adversarial Networks (GANs) for trajectory representation and synthetic data generation (Gao, et al., 2022) and Transformer-based models for advanced trajectory prediction (Xue & Salim, 2021). However, challenges persist due to the lack of unified tools and frameworks tailored for mobility data. For example, existing ML tools like TensorFlow and PyTorch lack native support for location-based data, complicating tasks such as clustering, classification, and similarity analysis. Developing foundational elements like mobility data embeddings could enhance model adaptability and lead to cohesive frameworks for mobility analytics (Vaswani, et al., 2017). Another challenge is the robustness of data-driven models in adapting to rapidly changing mobility patterns caused by events like the COVID-19 pandemic or societal shifts. Event-aware spatiotemporal networks have demonstrated potential in handling such scenarios (Wang, et al., 2022). Despite this, ensuring models remain resilient to evolving behaviors remains an open research area.

Behavioral understanding extends beyond traditional location prediction. Efforts aim to transition from predictive to prescriptive analytics, enabling actionable insights and policy-making. However, limitations such as a lack of labeled data and model explainability hinder progress. Techniques like disentangled representation learning (Zhao, Shao, Chan, & Salim, 2022) offer promise in improving explainability and addressing these challenges. Visualization and exploratory analysis are pivotal in mobility analytics (Andrienko, Andrienko, Bak, Keim, & Wrobel, 2013). Research combining modeling and simulation with visualization supports decision-making (Lee, et al., 2020). Yet, generalizing these approaches across domains and incorporating real-time human intelligence remains challenging. Integrating computational methods with human expertise could enhance understanding and modeling of traffic patterns, leading to better predictive and prescriptive analytics.

In this subsection, a detailed literature review of the vast field of traffic management and mobility science is presented. Moving on to the next subchapter, we focus on the background required for the task at hand which the present dissertation aims to address, namely automatic incident detection in intelligent transportation systems.

### *2.1.2 Classification of incidents*

Incidents are referring to "any […] event that causes a reduction of roadway capacity or an abnormal increase in demand" (Farradyne, 2000) . According to the authors in (Nikolaev, Sapego, Ivakhnenko, Mel'nikova, & Stroganov, 2017), incidents can be classified as planned or unplanned events. Figure 2-1 describes these events as mentioned in (Nikolaev, Sapego, Ivakhnenko, Mel'nikova, & Stroganov, 2017).


**Figure 2-1: Classification of incidents. (Nikolaev, Sapego, Ivakhnenko, Mel'nikova, & Stroganov, 2017)**

In reality, the classification of these events was done by considering the context of temporal, spatial, probability of occurrence and the cause of event (Amini, Papapanagiotou, & Busch, 2016). Incidents usually cause traffic disturbances such as a temporary reduction in capacity, "abnormal increase in traffic demand" (Beibei Ji, Jiang, Qu, & Chung, 2014), and fuel consumption. These negative impacts decrease the level of efficiency and safety of the road network. Therefore, early detection of incidents can be regarded as a required solution to facing them.

### *2.1.3 Traffic dynamics at the time of an incident*

It is important to understand the background and the basics of what happens in the traffic dynamics when an incident occurs. In this section, we first look at traffic fundamentals and how an accident impacts the dynamics of traffic observations and measurements. The fundamentals of traffic theory help us to better understand and interpret the input features and structure of data-driven models.

Traffic accidents are one of the important sources of traffic jams, and accidents cause a temporal local reduction of capacity. To explain the change in the traffic parameters, we need to look at the triangular fundamental diagram (Figure 2-2). The fundamental diagram of traffic flow represents the relation between the traffic features (i.e., flow(q), speed(u), and density(k)).

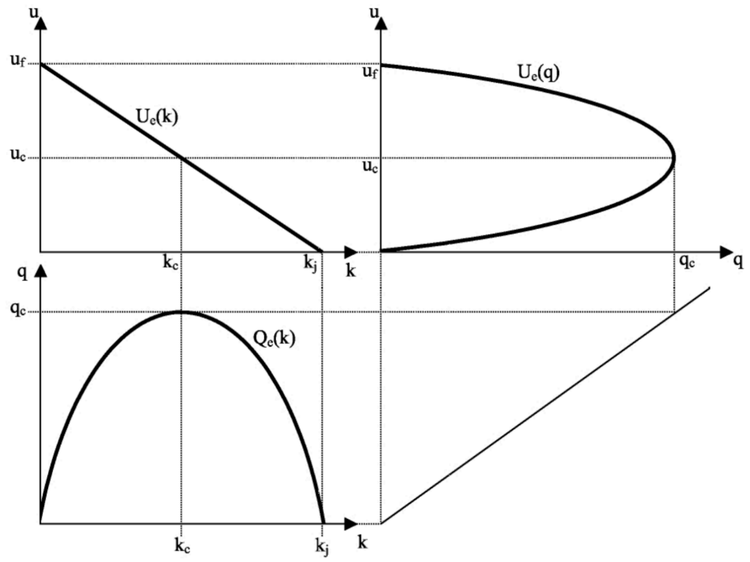



**Figure 2-2: The fundamental traffic diagrams according to Greenshield. (May., 1990)**

On the above diagram, *u* refers to the speed, *q* refers to the traffic flow whereas *k* refers to the density of the traffic.

As presented in Figure 2-3, when an accident occurs the traffic moves from uncongested state (point A) to congested state (point B). This change in the states affects the speed and flow of the vehicles. In other words, it is going to create a shockwave that will form a queue after the bottleneck (i.e., accident location). This phenomenon is often shown in the space-time diagram and will create a draw-up draw-down cycle in the speed-time graph.

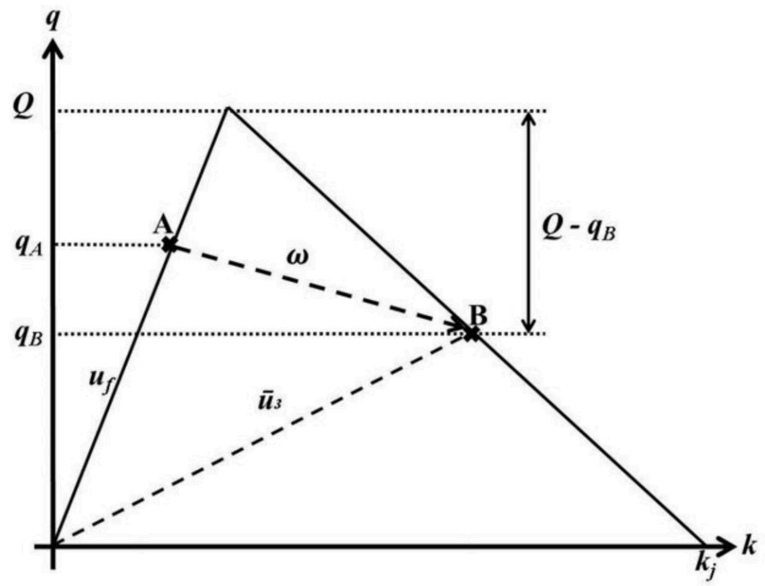



**Figure 2-3: Position of traffic states at the fundamental diagram when an accident occurs.**

Figure 2-4 illustrates the concept of shockwave and how the speed of the vehicles is going to change when the shockwave happens. In normal cases (i.e., non-accident), the traffic conditions do not vary significantly in sequences of time series between the upstream and downstream. On the other hand, traffic conditions between the upstream and downstream fluctuate rapidly when an accident occurs. This fluctuation is a result of the shockwaves caused by the accident. Mathematically, the speed of a shockwave (i.e., the speed at which congestion travels backward from the temporal bottleneck formed because of the accident) can be derived from the traffic characteristics (i.e., flow rate and density) of the upstream and downstream. Hence, the change in the speed dynamics when an accident occurs could be observed more significantly at the road sections after the accident location (Richards, 1956). To detect or predict an accident, one should look for the anomalies where the queue is formed (backward from the accident location). However, some anomalies may be observed in the upward direction as well. This information about the general dynamics of traffic at the time of an accident enhance our understanding of the anomaly points and how they should be interpreted.

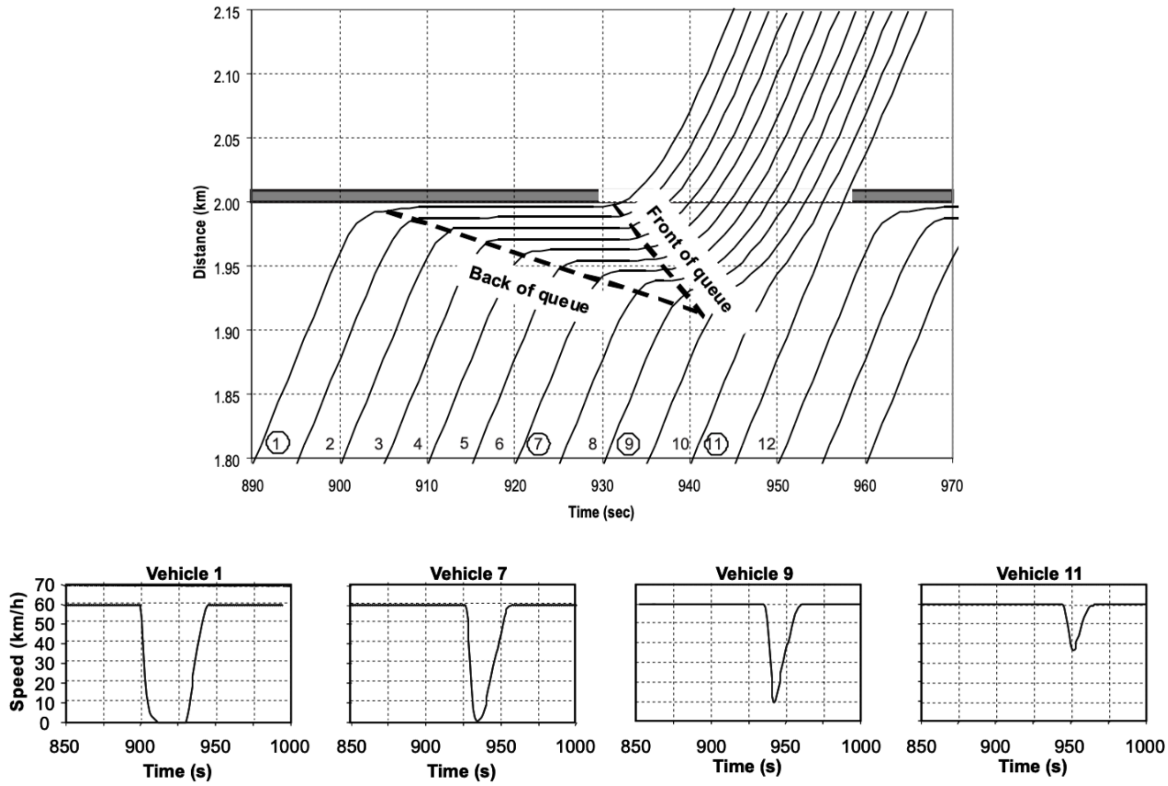



**Figure 2-4: An example of time–space diagram for typical temporary capacity reduction. (Francois Dion, 2004).**

## *2.2 Classification of Automatic Incident Detection Algorithms*

Traffic incident detection is a popular field in literature, since it is widely known that congestion in urban areas is often caused by traffic incidents. If such incidents could be detected in a timely manner, preventive measures could be rapidly taken. That is why in recent years, research efforts have been proposed to deploy Automatic Incident Detection (AID) Systems onto urban roads. In the following subsections, we present a classification of Incident Detection Algorithms (IDA) and go into some details regarding the significant differences between each group of algorithms.

Figure 2-5 presents the categories of Automatic Incident Detection Algorithms (AIDA), based on proposed approaches of various review papers, including (Hireche & Dennai, 2020; Li, Lin, Du, Yang, & Ran, 2022) (Evans, 2020) (Hireche & Dennai, 2020).

The AIDAs are grouped as comparative, time-series, and Artificial Intelligence (Statistical, Machine Learning and Deep Learning).

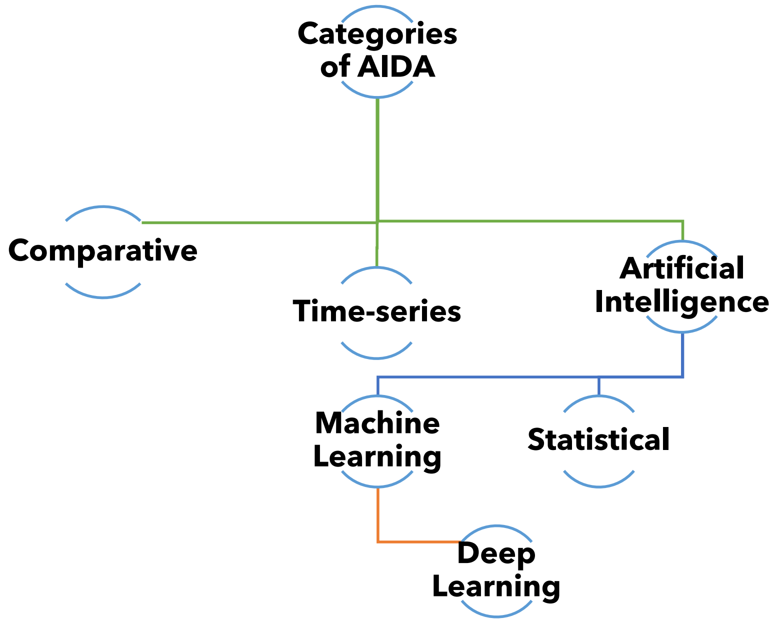



**Figure 2-5: Classification of Automatic Incident Detection Algorithms.**

In Table 2-1, some indicative studies are portrayed and grouped based on the proposed aforementioned classification.

**Table 2-1: Algorithms grouped by category and indicative works**

| *Category* | *Algorithm* | *Data attributes* | *Output (based on indicative works)* | *Indicative works* |
|---|---|---|---|---|
| *Comparative* | California | Occupancy from two adjacent detector stations | 2 states (0 incident-free; 1 incident) | (Payne & Tignor, 1978) |
| | California #7 | Occupancy from two adjacent detector stations | 4 states | (Balke, 1993) |
| | California #8 | Occupancy from two adjacent detector stations | 8 states | (Khoury, Haas, Mahmassani, & Logman, 2003) |
| *Timeseries* | ARIMA | Occupancy | Incident – No-incident: | (Ahmed & Cook, 1982) |

| | | | | |
|---|---|---|---|---|
| | | | An incident is detected if the observed occupancy value lies outside the confidence limits constructed two standard deviations away from the corresponding point forecasts. | |
| | Standard Normal Deviant (SND) | Occupancy | Incident – No-incident:<br><br>Compares 1-minute average occupancy measurements to archived occupancy values of the mean and SND defining the thresholds for detecting the incidents. | (Dudek, Messer, & Nuckles, 1974) |
| *AI*<br><br>*(Statistical and ML)* | Bayesian CNN | Occupancy, volume for incident and incident-free conditions, archived data on the type, location, and severity of incidents | Likelihood that an alarm is caused by incident. | (Liu, Jin, Li, Hu, & Lia, 2022) (Zhu, Guo, Krishnan, & Polak., 2018) |
| | SVM | Speed, flow, occupancy and derived features | Incident – No-incident | (Li, Hu, X., & Zhou, 2017) (Dardor, Chlyah, & Boumhidi, 2018) |
| | Neural Networks | Volume, speed, occupancy and derived features | Incident – No-incident | (Shang, Feng, & Gao, 2020) (Zhu, Guo, Krishnan, & Polak., |

| | | | |
|---|---|---|---|
| | | | 2018) (Zhu, Wang, Yan, Guo, & Tian, 2022) |
| Wavelet Transformation with Logistic Regression | Volume, speed, occupancy and derived features | Probability of incident | (Agarwal, Kachroo, & Regentova, 2016) |
| Isolation Forest | Volume, speed, occupancy and derived features | Incident – No-incident | (Zhu, Wang, Yan, Guo, & Tian, 2022) |
| GANs | Volume, speed, occupancy and derived features | Incident – No-incident | (Li, et al., 2019) (Lin, Liu, Li, & Qu, 2023) |
| LSTM | Volume, speed, occupancy and derived features | Incident – No-incident | (Cui, Ke, & Wang, 2018) (Zhu, Wang, Yan, Guo, & Tian, 2022) |
| Graph Neural Networks | Volume, speed, occupancy, graph traffic network and derived features | Incident – No-incident | (Zhou, Wang, Xie, Chen, & Liu, 2020) (Yu, et al., 2021) (Wang, Lin, Guo, & Wan, 2021) |

In the following subsections, some of the representative works for each category type are presented and discussed.

### 2.2.1 *Comparative Algorithms*

The popular California and McMaster algorithms are representative of this type of model and have been widely applied (Hall, Shi, & Atala, 1993).The California algorithms are amongst the most commonly used and replicated IDAs. Many variations of the original have been presented and compared (Payne & Tignor, 1978), but all of them use pre-set decision trees based on traffic variables, to classify real-time traffic conditions into incident and non-incident states. Because of their simplicity, many studies have used the California algorithms as a benchmark for comparison, and many others have iterated on the first version presented to improve its performance and limit its drawbacks.

However, these simple models cannot provide sufficient accuracy to meet the requirements of an Intelligent Transportation and AID System (Samant & Adeli, 2000).

### 2.2.2 *Time series Algorithms*

One of the earliest and simplest AIDAs was the standard normal deviate (SND) algorithm (Dudek, Messer, & Nuckles, 1974) .The algorithm was developed for motorways, and used occupancy data to detect the "shock wave" (i.e. sudden change to lower speeds) in traffic caused by incidents. (Dudek, Messer, & Nuckles, 1974) tested a number of different values of parameters, but occupancy was found to produce the best results. This method detected abnormally high values of occupancy, which would indicate queuing traffic, which would indicate the occurrence of an incident. However, as stated in (Dudek, Messer, & Nuckles, 1974), although the 1.3% false alert rate appears low, "the number of false alarms can become very significant in an operational system". This high rate may be because the IDA only uses occupancy values, and so can only detect congestion, rather than differentiating incidents. It would also not be able to detect incidents upstream of detectors (where low flows may occur), and did not consider spatial patterns to detect incidents (e.g., nearby detectors raising alerts which raise likelihood of an incident occurring).

Auto-Regressive Integrated Moving-Average time series (ARIMA) models use recent observations of a selected traffic variable to create a prediction of its "expected" value (i.e., conditions that would occur if no incident occurred) in the near-term future (Ahmed & Cook, 1982). If real-time values significantly deviate from this prediction, an incident alert is raised. For instance, ARIMA was used to detect incidents using occupancy data on freeways in Detroit, U.S.A. ARIMA models are commonly found to be effective in forecasting traffic variables in the short-term future. However, the forecast would not be of "expected" traffic conditions if the recent observations are influenced by incidents. When used in an AIDA, this could lead to incidents going undetected. The model is also known to be less effective during sudden changes in traffic parameters, for instance during rush hour.

(Thancanamootoo & Bell, 1988) presented one of the first AIDAs designed specifically for urban networks. It used volume and occupancy data to detect incidents between pairs of upstream/downstream detectors. (Sheu & Ritchie, 1998) presented a modified sequential probability ratio tests algorithm for use in urban networks. It included three procedures, a knowledge-based rule set for identifying the symptoms of an incident, signal processing for real-time prediction of incident-related traffic conditions and pattern recognition for incident detection. (Lee & Taylor, 1999) also detected incidents on urban streets, but by applying a Kalman filtering algorithm to find sudden changes in traffic variables on a two-lane arterial's detectors. This approach was designed to be simple and require little calibration, while being dynamic enough to account for traffic signals.

The most crucial factor which makes time-series differ from comparative algorithms is the fact that the threshold used to raise incidents varies based on recent local conditions. This gives an advantage because it means temporal variations (such as peak periods) can be more readily accounted for automatically, and less manual calibration is required. (Evans, 2020)

### *2.2.3 Artificial Intelligence Algorithms*

New challenges to the research community have been introduced regarding the enhancement of the performance of AID systems. To overcome these challenges, and to improve the efficiency and safety of road traffic, attention has been drawn to the use of Artificial Intelligence (AI) techniques. This section covers a review of the key machine learning techniques used in literature, including Statistical-Based and Machine Learning based techniques. The ML techniques include Support Vector Machines (SVMs), Neural Networks (NNs), Generative Adversarial NNs, Graph NNs, Decision Trees (DT), Naïve Bayes (NB), Autoencoders, Long Short-Term Memory (LSTM) networks and Ensemble Learning (EL).

#### 2.2.3.1 Statistical-based Algorithms

These types of models test differences in traffic flows based on statistical techniques, where a significant difference indicates a possible incident. To capture the temporal and spatial correlations among traffic flows, some studies implemented advanced statistical techniques. For example, an autoregressive integrated moving average model was built to detect traffic incidents on the Lodge Highway in Detroit; (Ahmed & Cook, 1979) the proposed detection logic performed smoothing using a moving average filter and obtained better results (Chassiakos & Stephanedes, 1993). Later, a multiple model particle smoother was introduced to convert the incident detection problem into a traffic state prediction problem and solve it effectively (Wang, Fan, & Work., 2016). Although statistics-based models have been widely applied, they have some shortcomings. First, the algorithm assumptions may not be consistent with the actual traffic flow data. Second, these models are highly dependent on user experience. When implementing a statistics-based model, the thresholds are often set manually by the users. Moreover, these models sometimes cannot simultaneously consider the temporal and spatial correlations among traffic flow data (Li, et al., 2019)

#### 2.2.3.2 Machine Learning and Deep Learning algorithms

Figure 2-6 shows the taxonomy of all ML techniques used in the AID systems of examined studies. For a more detailed description of the reviewed papers, we encourage the readers to consult the respective review paper (Hireche & Dennai, 2020).

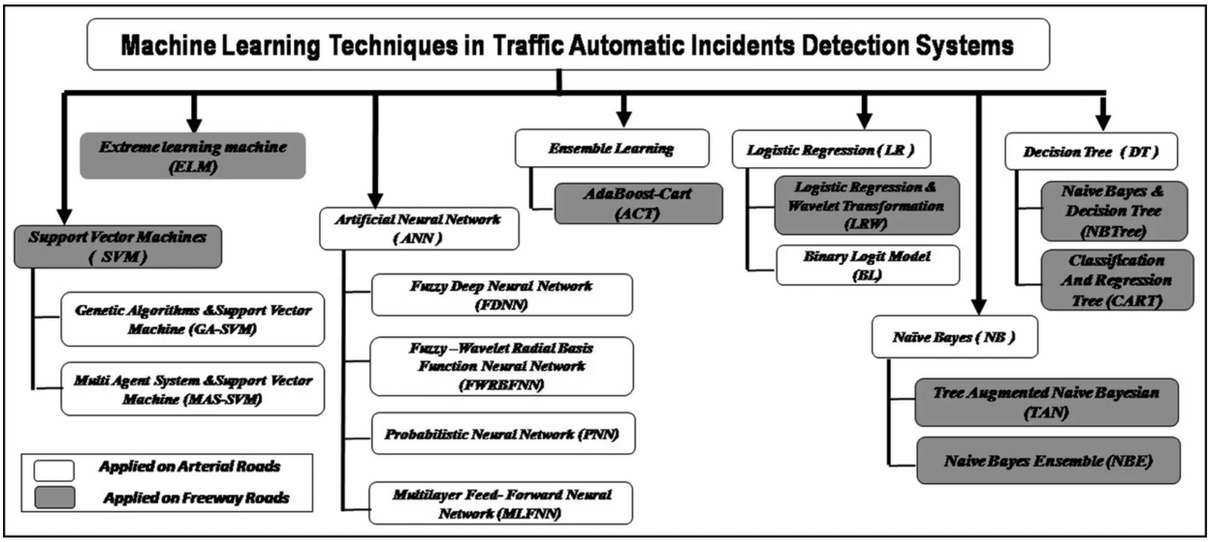



**Figure 2-6: Taxonomy of machine learning techniques used in traffic automatic incident detection (Hireche & Dennai, 2020)**

To make the incident detection model more flexible and robust, various machine learning models have been applied. The traffic incident detection problem is first converted into a binary classification task in which an incident is defined as a "1" and a non-incident is defined as a "0". Then, a machine learning model such as a Support Vector Machine (SVM) (Yuan & Cheu., 2003) (Xiao & Liu., 2012), Classification Tree (CT) (Chen & Wang, 2009), Random Forest (RF) (Liu, Jian Lu, & Chen., 2013) or neural network (NN) (Samant & Adeli, 2000) can be used to solve the task. Li et al. compared some famous machine learning models and found that ensemble approaches improve the performance. Adding a bagging strategy for instance to an SVM increases the accuracy (Li, He, Zhang, & Yang.., 2016).

Some advanced NN models have been widely applied in previous traffic incident detection studies and have obtained very good results. Ma et al. used a deep neural network to recognize traffic congestion on a highway network using both temporal and spatial traffic flow characteristics (Ma, Yu, Wang, & Wang., 2015). Zhu et al. developed an incident detection model at the network level based on a Convolutional Neural Network (CNN) (Zhu, Guo, Krishnan, & Polak., 2018). Moreover, in their study, Almohammad & Georgakis focused on how predicted incidents can be simulated in order to predict their impacts on the transport network performance. A method has been suggested to convert real traffic events into simulation incidents, consisting of two main components; machine learning based model for event classification and event-to-incident mapping. Various algorithms were evaluated, modeling diverse real-world events (Almohammad & Georgakis, Machine Learning Based Method for Modeling Traffic Events, 2022). It has been proven that Deep Learning (DL) models outperform traditional machine learning models because they can fully mine the traffic information from the data. However, achieving a sufficient number of samples is difficult when applying a deep learning model. Consequently, simulated data have been widely used, but sometimes such data does not represent the true highway traffic flow. (Lv, Duan, Kang, Li, & Wang, 2015) (Ma, et al., 2017) (Zhu, Guo, Krishnan, & Polak., 2018). Another method applied to solve the small sample size problem is to only collect samples during each incident as incident samples, in order to increase the sample size. However, this approach could affect the real-time capacity of the model. (Li, Lin, Du, Yang, & Ran, 2022)

An important study presented by Li et al. compared four classification methods to detect traffic incidents. These methods included SVM, NB, Cart, and AdaBoost-Cart (ACT). (Li, Hu, X., & Zhou, 2017)After evaluating these classification methods, the results indicated that AdaBoost-Cart and NB models performed quite well. In 2018, Dardor et al.  tried to resolve the problem of incident detection on signalized intersection urban areas based on SVM coupled with Genetic Algorithm (GA-SVM) model (Dardor, Chlyah, & Boumhidi, 2018)In this proposition, the Radial Basis Function (RBF) was selected as the kernel function of SVM to classify the signal and determine the event type, while GA is selected as the optimization algorithm to maximize classification accuracy of SVM. SVM models can provide faster results and a lot of customization options. In addition, SVM require less computational cost, which is vital for real-time incident detection.

It is worth mentioning that recent studies have used big data collected from social media platform streams. Specifically, in two distinct studies, authors use twitter data to identify anomalies in the network and signal those potential disruptions to affected stakeholders. In the first paper, the authors present a methodology for real-time traffic event detection using geolocated tweets. Tweets are processed with natural language techniques and classified to identify traffic-related content. Applied in the West Midlands, UK, the approach achieved a considerable accuracy (92.86%) (Jones, Georgakis, Petalas, & Suresh, 2018). The second paper examines the use of geolocated Twitter data to predict transport network conditions, such as disruptions or congestion, in Greater Manchester. By analyzing the relationship between actual network status and synthesized data from tweets, it addresses whether tweet sentiments near incident areas differ from those in normal traffic zones, using sentiment analysis techniques. (Almohammad & Georgakis, Public Twitter Data and Transport Network Status, 2020)

Generative Adversarial Networks have also recently been used for anomaly detection for spatiotemporal events. (Li, et al., 2019) proposed MADGAN, an unsupervised anomaly detection method for multivariate time series based on GAN. They trained a GAN generator and discriminator with LSTM. Then, the GAN-trained generator and discriminator are employed to detect anomalies in the testing data with a combined Discrimination and Reconstruction Anomaly Score (DR-Score).

Furthermore, Recurrent Neural Networks (RNNs) show promise to work well with sequential data like time-series. They have also been leveraged for traffic accident prediction thanks to their generally high performance and the availability of time-series data (Wang & Abdel-Aty, 2006.). For example, (Ren, Song, Liu, Hu, & Lei., 2017) proposed a deep learning approach (RNN) to predict traffic accident risk, where risk is defined as the number of accidents in a region at a certain time. (Chen, Song, Yamada, & Shibasaki, 2016) used a similar concept of traffic accident risk and developed an Autoencoder deep architecture to understand the impact of human mobility on traffic accident risk.

Last but not least, Graph Neural Networks (GNNs) ( (Li, Yu, Shahabi, & Liu, 2017), (Lin, Zhengbing, Srinivas, & Peeta, 2018), (Cui, Henrickson, Ke, & Wang, 2019), (Cui Z. , Ke, Pu, Ma, & Wang, 2020)) have gained interest to address the complexities of traffic prediction and incident detection, leveraging the inherent graph structure of traffic networks. Notably, Zhou et al. (2020) introduced a model known as the Differential Time-varying Graph neural network (DTGN), designed to detect real-time traffic shifts and the dynamic connections between different areas within a traffic graph. (Zhou, Wang, Xie, Chen, & Liu, 2020). This approach notably refines predictions to a minute-by-minute basis and isolates the urban areas most prone to accidents. In 2021, Yu et al. developed a Graph Convolutional Network (GCN) tailored for predicting road incidents by assimilating both spatial-temporal and external data within a graph-based representation of traffic flows. (Yu, et al., 2021) Following this, Wang et al. introduced the GSNet model in the same year, aiming to understand the spatio-temporal patterns and relationships across different regions by analyzing geographic and semantic data integrated from undirected graphs, which embody various road network characteristics. (Wang, Lin, Guo, & Wan, 2021)

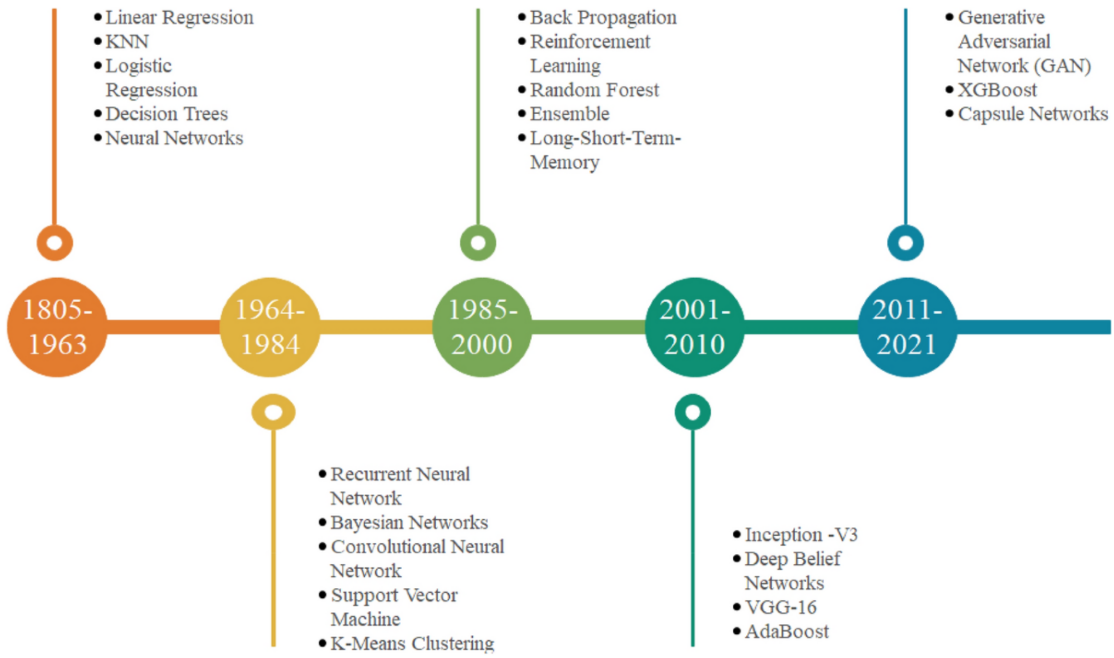



**Figure 2-7: The evolution of methods in Automatic Incident Detection over time. (Ahsan & Siddique, 2021)**

Based on the aforementioned works, it becomes evident that the evolution of techniques in automatic incident detection mirrors the broader advancements in machine learning and artificial intelligence over time as shown in Figure 2-7. In the early days (1805-1963), foundational methods such as linear regression and logistic regression laid the groundwork. Moving into 1964-1984, statistical and simpler machine learning techniques like decision trees and early neural networks emerged, improving the ability to analyze and predict incidents. The period from 1985 to 2000 saw the introduction of more complex models like recurrent and convolutional neural networks, along with support vector machines, enhancing the capacity to handle larger datasets and more intricate patterns. Between 2001 and 2010, advancements such as ensemble methods, random forests, and the introduction of long-short-term memory (LSTM) networks allowed for deeper analysis and better temporal understanding in incident prediction. In the most recent decade (2011 - today), innovative techniques such as generative adversarial networks (GANs), XGBoost and more complex neural networks have further refined detection capabilities, enabling more sophisticated, real-time, and accurate incident prediction systems. This progressive development of algorithms reflects the field's growing ability to anticipate and manage incidents in complex urban environments.

## *2.3 Related EU Projects and Schemes around the globe*

In this section, some related EU-funded initiatives and projects focused on automatic incident detection and the improvement of urban environments, utilizing AI and other advanced technologies are presented:

1. **TANGENT**[1]: The TANGENT project, funded under the European Commission's Research and Innovation Programme, aims to develop tools for optimizing traffic operations in a coordinated and dynamic way. It focuses on multimodal transport management, integrating both automated and non-automated vehicles, passengers, and freight transport. The project includes enhanced traffic information services, real-time traffic management services, and transport network optimization across various cities such as Rennes, Lisbon, Greater Manchester, and Athens.

2. **ORCHESTRA**[2]: The ORCHESTRA project aims to connect services to make mobility and logistics run smoothly to cope with diverse demands and situations across transport modes. The ORCHESTRA project managed to establish a common understanding of multimodal traffic management concepts and solutions, with and across modes, for various stakeholders and multiple contexts. It defined a Multimodal Traffic Management Ecosystem (MTME) where traffic managements in different modes and areas (rural and urban) are coordinated to contribute to a more balanced and resilient transport system, bridging current barriers and silos.

[1] https://tangent-h2020.eu/
[2] https://orchestra2020.eu/

3. **CIVITAS Initiative[3]:** CIVITAS is one of the key European Union initiatives aiming to support cities in implementing sustainable urban mobility measures. Since its inception in 2002, CIVITAS has funded over 80 cities in developing smart transport systems, including traffic management and automatic incident detection solutions. Through various demonstration projects, the initiative has piloted the use of AI and IoT technologies for real-time traffic monitoring and congestion management, thus contributing to the overall safety and efficiency of urban transport networks.

4. **CONDUCTOR[4]:** The CONDUCTOR project aims to design and demonstrate advanced traffic and fleet management systems that prioritize seamless multimodality, efficient transportation of passengers and goods, and interoperability between automated and conventional vehicles. Key objectives include dynamic load balancing, integrating ride-parcel pooling, and optimizing multi-modal systems to enhance traffic management. The project is part of the EU’s efforts to develop connected, cooperative, and automated mobility (CCAM), focusing on resilience and sustainable transport solutions for future cities.

5. **DELPHI[5]**: The DELPHI project aims to integrate passenger and freight transport into a unified, federated system for efficient, multimodal mobility. It leverages AI, machine learning, and advanced monitoring technologies, such as unmanned aerial systems, to optimize transport flows and data sharing across urban, suburban, and rural networks. The project will conduct pilot demonstrations in Spain, Greece, and Romania to test these innovative systems.

6. **ACUMEN[6]**: The ACUMEN project aims to facilitate seamless, sustainable, and safe door-to-door journeys for both people and goods by creating a dynamic,

[3] https://civitas.eu/
[4] https://conductor-project.eu/
[5] https://delphi-project.eu/
[6] https://acumen-project.eu/

AI-driven framework for multimodal traffic management. It focuses on improving network efficiency and shared mobility options, which reduces travel costs and increases the uptake of sustainable transport solutions. The project integrates advanced data sharing and decision-making tools to enhance overall system performance and manage traffic more effectively across multiple cities.

## 2.4 Conclusions and Research gaps

From the algorithms that have been used in literature, it is demonstrated that Machine Learning AIDAs state some of the best results in terms of the domain established evaluation metrics, such as false alert rates. Such algorithms aim to learn the conditions of an incident and so may be able to differentiate incidents from context. Moreover, transport simulators provide simplified versions of real-world networks, which often do not account for real-world disruptions such as emergency vehicles passing at high speed, erratic driving, or major sporting events. Hence, transport simulators typically output more predictable traffic data values, meaning AIDAs can perform better. As such, it is unclear whether results on simulated data are replicable on field data, and if so, how much calibration would be required. From studies of those implemented in the field, traffic variable based AIDAs appear well suited to motorways and arterials but find difficulty in accounting for traffic signal noise and contexts within urban streets and junctions.

Based on the aforementioned research works and reviews on the matter, it is shown that incident detection is still far from being a resolved problem. Despite progress being made throughout the years, state of the art IDAs are found to still have outstanding limitations and challenges, some of which are detailed below:

- **Quality of Collected Data:** A critical challenge in automatic incident detection lies in the inherent inaccuracies of mobility data. GPS coordinates are often prone to errors, and low sampling rates can introduce temporal gaps in data collection, resulting in uncertainty about the exact movement of vehicles or pedestrians between data points. Although methods like map matching and interpolation have been employed to mitigate these issues, they often rely heavily on existing infrastructure (e.g., road networks), which may itself contain inaccuracies. Developing scalable, fine-grained models capable of handling city-scale datasets without relying on such infrastructure remains a challenge.

- **Explainability of Machine Learning Models:** One of the major challenges in deploying ML for incident detection is the lack of interpretability in models, particularly with deep learning systems. These models often operate as "black boxes," making it difficult to understand how and why certain predictions are made. For safety-critical applications like traffic incident detection, this opacity can be problematic. Explainability is essential for gaining stakeholder trust, and explainable approaches offer a pathway to making models more interpretable by isolating and explaining underlying factors that influence decisions.
- **Handling Spatiotemporal Complexity**: Mobility data introduces the complexity of both spatial and temporal dimensions, making it difficult to apply traditional ML techniques that are not designed for such multi-dimensional data. Proximity in space and time plays a crucial role in incident detection, but many ML models struggle to integrate these factors effectively. Handling spatiotemporal data streams in real-time while accounting for the constantly evolving nature of traffic patterns and incidents is a significant technical difficulty.
- **Privacy Concerns**: Given that mobility data is considered sensitive information, there are significant privacy challenges when using this data for automatic incident detection. Existing anonymization techniques for traditional data are often inadequate for mobility data, where location trajectories can inadvertently reveal personal details such as home addresses, workplaces, or even behaviors. Ensuring privacy while maintaining the utility of the data for ML-based incident detection is an ongoing challenge.

- **Human-in-the-Loop Systems**: While fully automated systems for incident detection are desirable, there is a recognized need for human oversight in real-time, high-stakes scenarios. However, integrating human expertise into ML workflows presents its own set of challenges, including how to involve human decision-makers without slowing down the system's response time. Current systems often rely heavily on automated processes, in general, limiting the ability to incorporate human intelligence in real-time analysis. Striking the right balance between automated systems and human intervention is a key challenge.

From the identified challenges, several future directions emerge:

- **Advancements in Data Pre-processing Techniques:** As mobility data continues to grow in volume and complexity, there is an urgent need for improved data cleaning techniques. Future research should focus on developing more robust and scalable methods for handling data inaccuracies, such as trajectory interpolation, that can operate in real-time and across various types of environments (urban, highway etc.). Additionally, research should explore ways to address gaps in temporal data without relying heavily on existing infrastructure, as this will improve the reliability of incident detection systems.
- **Improving Explainability in ML Models**: To increase the adoption and reliability of ML systems for automatic incident detection, future research should prioritize improving the explainability of these models. One promising direction is the development of techniques which can isolate the key spatiotemporal factors driving model decisions. Furthermore, integrating interpretable models into workflows that involve multiple stakeholders (such as city planners and emergency responders) can enhance trust and ensure that the systems are understood and validated by human experts.

- **Spatiotemporal ML Models**: The development of specialized machine learning models that can effectively process spatiotemporal data is another promising research avenue. Existing models, originally designed for tasks like image recognition or language processing, are not well-suited to handling the complexities of traffic data, where both location and time are critical. Researchers should focus on developing models that natively understand the importance of spatiotemporal relationships, enabling more accurate incident detection. Incorporating real-time data streams and continuously evolving traffic patterns into these models will be essential.
- **Enhanced Privacy-Preserving Techniques**: As mobility data becomes more ubiquitous, ensuring privacy will remain a key concern. Future research should focus on developing advanced privacy-preserving mechanisms that allow for the collection and analysis of sensitive mobility data without compromising individual privacy. Approaches such as geo-indistinguishability, which protect user identities while maintaining data utility, are likely to become more widely adopted.
- **Human-in-the-Loop Systems for Enhanced Decision Making**: Future incident detection systems should aim for more effective integration of human expertise into real-time analysis workflows. This could be achieved through the development of hybrid systems that combine ML algorithms with human reasoning, particularly in complex scenarios where automated systems may struggle. By leveraging human expertise in real-time, such systems can provide more reliable and context-aware responses to incidents. Research into optimizing these interactions without sacrificing system efficiency will be crucial.

# 3 Research Challenges

In the current Chapter, the research questions along with the Thesis outline, are presented in detail. More precisely, the research questions are reviewed and divided into their parameters on which the Thesis is positioned. Besides, a synopsis of the propositions of the Thesis is also provided.

## 3.1 Research Questions

In this Section, the research questions of the current Thesis are displayed. The summarized Table 3-1 states the research challenges the Thesis aims to provide answers to, along with the corresponding parameters which they consist of.

**Table 3-1: Research Questions and the corresponding Parameters**

| Research questions | Parameters |
|---|---|
| What are the key components and methodologies for real-time monitoring and prediction in AI-based traffic incident detection? | • What are the characteristics of traffic in case of an incident?<br>• What are the essential data sources for real-time traffic monitoring and incident detection?<br>• What are the main categories of algorithms for the incident detection task? What are the strong and weak points of each category?<br>• Which AI algorithms are most effective and have been thoroughly proposed by the literature for traffic incident detection? What are the advantages and limitations of each?<br>• What performance metrics are critical for evaluating the effectiveness of traffic incident detection systems? |
| How can human-centered traditional and automated AI technologies be | • What are the steps for building a comprehensive framework/methodology using AI for real-time detection of planned and unplanned incidents? |

| | |
|---|---|
| leveraged to develop a comprehensive framework for real-time detection of traffic incidents and monitoring of urban networks? | • How can human expertise be integrated into AI-based traffic monitoring and incident detection systems to build a more reliable and trustworthy incident detection system? |
| How do AI-driven methodologies and algorithms enhance the detection of planned and unplanned traffic incidents? | • What are the primary advantages of using AI for traffic incident detection compared to traditional methods?<br>• What limitations or challenges remain in the current AI-driven approaches?<br>• How do different data-driven, machine learning and deep learning models perform in the context of traffic incident detection? Are there differences in the techniques employed in detecting, on the one hand, planned and, on the other hand, unplanned incidents?<br>• What are the key features and parameters that influence the effectiveness of the AI models? |
| How can AutoML techniques enhance the development of AI models for traffic incident detection? | • What is Automated Machine Learning, and could it have a role in the context of Intelligent Transportation Systems?<br>• Which AutoML frameworks, methods and tools are the most suitable for urban incident detection?<br>• How can specific stages of model building such as model selection and optimization be automated effectively?<br>• How do AutoML techniques compare to traditional methods regarding their performance? |
| How to ensure predictions of AI-based traffic incident detection systems are explainable and trustworthy while | • What is the role of human feedback and how it can be leveraged in AI-based systems?<br>• What explainability techniques can be used to make AI predictions understandable and what tools can be integrated in such systems?<br>• What mechanisms can be used to incorporate expert feedback using a human-in-the-loop approach into AI-based intelligent |

| | |
|---|---|
| integrating expert feedback? | transportation systems to enhance the quality of the predictions?<br>• What are the results of integrating human in the loop in terms of performance when retraining the ML models? Do explainability techniques have an impact on the trustworthiness of the system? |

### *3.1.1 Research Question 1: What are the key components and methodologies for real-time monitoring and prediction in AI-based traffic incident detection?*

In the context of developing AI-based traffic incident detection systems, it is crucial to identify and understand the key components and methodologies for real-time monitoring and prediction. This question aims to present the basic elements and methods used in literature and state-of-the-art works in the mobility field, initially, and particularly focus on the incident detection task, thus providing a comprehensive understanding of used technologies and techniques.

The aforementioned question shapes Research Question 1 (RQ1) of the Thesis and is further divided into smaller sub-questions. Concretely, RQ1 can be split upon the following points:

- What are the characteristics of traffic in case of an incident?
- What are the essential data sources for real-time traffic monitoring and incident detection?
- What are the main categories of algorithms for the incident detection task? What are the strong and weak points of each category?
- Which AI algorithms are most effective and have been thoroughly proposed by the literature for traffic incident detection? What are the advantages and limitations of each?
- What performance metrics are critical for evaluating the effectiveness of traffic incident detection systems?

These points are discussed and answered in Chapter 2. The current Ph.D. thesis analyses the main objectives of data-driven and AI-based traffic incident detection. Section 2.2, focusing on the state-of-the-art works in the domain of automatic incident detection and real-time monitoring and prediction systems. The Thesis also aspires to provide insights by synthesizing the current practices, identify challenges and gaps and propose future directions.

### *3.1.2 Research Question 2: How can human-centered traditional and automated AI technologies be leveraged to develop a comprehensive framework for real-time detection of traffic incidents, monitoring, and situational awareness of urban networks?*

Integrating human-centered traditional and automated AI technologies is essential in order to create a comprehensive framework that facilitates real-time detection, monitoring, and situational awareness of traffic incidents in urban networks.

The Research Question 2 (RQ2) of the current Thesis, as stated above, is further divided into smaller sub-questions, which are the following:

- What are the steps for building a comprehensive framework/methodology using AI for real-time detection of planned and unplanned incidents?
- How can human expertise be integrated into AI-based traffic monitoring and incident detection systems to build a more reliable and trustworthy incident detection system?

This Ph.D. thesis aims to investigate the integration of human-centered traditional and automated AI technologies as part of a comprehensive framework and the analysis conducted and suggested propositions are presented in Chapter 4. The primary focus of this Chapter is to describe a robust and innovative framework for enhancing real-time traffic incident detection and management in addition to the steps required as part of a comprehensive methodology, with the aim of improving urban traffic conditions and situational awareness.

### *3.1.3 Research Question 3: How do AI-driven methodologies and algorithms enhance the detection of planned and unplanned traffic incidents?*

Based on the extensive literature review conducted as part of Chapter 2, it becomes apparent that AI-driven methodologies have significantly contributed to the field of traffic incident detection by offering advanced capabilities to identify incidents promptly and in some cases, more efficiently than traditional methods. This research question aims to present the methods, tools and algorithms used in this context for both planned and unplanned incidents.

The primary research question (RQ3) is divided into the following sub-questions to comprehensively address the topic:

- What are the primary advantages of using AI for traffic incident detection compared to traditional methods?
- What limitations or challenges remain in the current AI-driven approaches?
- How do different data-driven, machine learning and deep learning models perform in the context of traffic incident detection? Are there differences in the techniques employed in detecting, on the one hand, planned and, on the other hand, unplanned incidents?
- What are the key features and parameters that influence the effectiveness of the AI models?

These points are thoroughly explored and discussed in Chapter 5. The current Ph.D. thesis aims to analyze the use of AI-driven and advanced analytics techniques in traffic incident detection. Thus, the advantages and challenges of these methods are described. By answering the aforementioned sub-questions, this research offers insights into the capabilities and limitations of AI and the use of advanced analytics in detecting planned and unplanned traffic incidents.

### *3.1.4 Research Question 4: How can AutoML techniques enhance the development of AI models for traffic incident detection?*

AutoML techniques offer significant potential for enhancing the development of AI models by automating various aspects of the model-building process. This research question investigates how these techniques could be applied to traffic incident detection to improve model performance and streamline the development process. RQ4 includes the following points:

- What is Automated Machine Learning, and could it have a role in the context of Intelligent Transportation Systems?
- Which AutoML frameworks, methods and tools are the most suitable for urban incident detection?
- How can specific stages of model building such as model selection and optimization be automated effectively?
  How do AutoML techniques compare to traditional methods regarding their performance?

These points are thoroughly explored and discussed in Chapter 6. The current Ph.D. dissertation analyzes the use of AutoML techniques in the development of AI models for traffic incident detection and focuses on evaluating the effectiveness of AutoML in automating model development and improving performance. Therefore, the thesis provides valuable insights into the capabilities of AutoML and its application in traffic management systems.

### *3.1.5 Research Question 5: How to ensure human in the loop and prediction is explainable and transparent in AI-based traffic incident detection systems?*

Ensuring that AI-based traffic incident detection systems are explainable, transparent, and involve human oversight is crucial for gaining user trust and maintaining high performance while adhering to ethical standards. This question explores methods and practices to make AI predictions understandable and transparent while incorporating human judgment and expertise in urban traffic systems.

RQ5 is divided in the following sub-questions, more specifically:

- What is the role of human feedback and how it can be leveraged in AI-based systems?
- What explainability techniques can be used to make AI predictions understandable and what tools can be integrated in such systems?
- What mechanisms can be used to incorporate expert feedback using a human-in-the-loop approach into AI-based intelligent transportation systems to enhance the quality of the predictions?
- What are the results of integrating human in the loop in terms of performance when retraining the ML models? Do explainability techniques have an impact on the trustworthiness of the system?

These points are discussed in Chapter 7, where the integration of human feedback (human-in-the-loop approaches) and explainability techniques in AI-based traffic incident detection systems are discussed in detail.

## *3.2 The Propositions of the Thesis*

The solutions this Thesis suggests intend to cover the research gaps that are identified and hence respond to the Research Questions stated above. Table 3-2 below displays how the various articulated research questions have been associated with relative scientific publications produced during this study, as well as the Chapters in which the insights of the research are described (Chapters 4, 5, 6, 7). Chapter 8 discusses the system development and Chapter 9 presents the evaluation of approaches presented in Chapters 4, 5, 6 and 7 in two real-world case studies.

**Table 3-2: Positioning of the Thesis Proposition following the Research Questions.**

| Research Questions | Thesis Proposition | Related Publications | Chapter |
|---|---|---|---|
| What are the key components and methodologies for real-time monitoring and prediction in AI-based traffic incident detection? | • Analysis of methodologies as proposed by literature<br>• Synthesis of literature review and research gaps identification | [j2] [c2] [c3] | 2 |
| How can human-centered traditional and automated AI technologies be leveraged to develop a comprehensive framework for real-time detection of traffic incidents and monitoring of urban networks? | • Overall framework for comprehensive incident (planned and unplanned) detection | [j3] [c1] | 4 |

| | | | |
|---|---|---|---|
| How do AI-driven methodologies and algorithms enhance the detection of planned and unplanned traffic incidents? | • Comprehensive discussion about ML and DL algorithms and their advantages and challenges<br>• AI-based and Advanced Analytics Methodology to identify planned and unplanned traffic incidents<br>• Comparison of state-of-the-art algorithms | [j2] [j3] [c2] [c3] | 5 |
| How can AutoML techniques enhance the development of AI models for traffic incident detection? | • Novel AutoML-based methodology in detecting unplanned traffic incidents with data pre-processing pipeline<br>• Contrast of AutoML-based approach with General Approach Algorithms<br>• Guidelines and best practices in integrating autoML in overall framework | [j2] | 6 |
| How to ensure predictions of AI-based traffic incident detection systems are explainable and trustworthy while integrating expert feedback? | • Interactive Feedback from operators to validate prediction<br>• Enhanced dataset quality through functionality of manual incident insertion<br>• Explainability features in AI-Based Traffic Incident Detection Systems | [c1] [c4] | 7 |

# 4 Framework for Real-Time Monitoring and Prediction of Traffic Incidents

This Chapter presents the comprehensive framework proposed in the area of real-time monitoring and prediction in automatic traffic incident detection. The focus is placed on integrating both traditional and automated AI technologies to enhance urban traffic management systems. Research has been conducted to identify and evaluate key components, methodologies, and techniques essential for effective traffic incident detection and traffic monitoring. This includes investigating the integration of human-centered approaches with traditional and automated approaches utilizing advanced data analytics to improve the overall effectiveness, reliability and trustworthiness of the system.

## *4.1 Pillars of our framework*

The proposed framework for an advanced incident detection system is based upon four pillars: AI and Data Analytics, Automated Machine Learning, Explainability and Human-in-the-Loop. Each pillar represents a critical component of the system, contributing to its robustness. These pillars are illustrated and briefly discussed in Figure 4-1.

Four pillars of our proposed framework

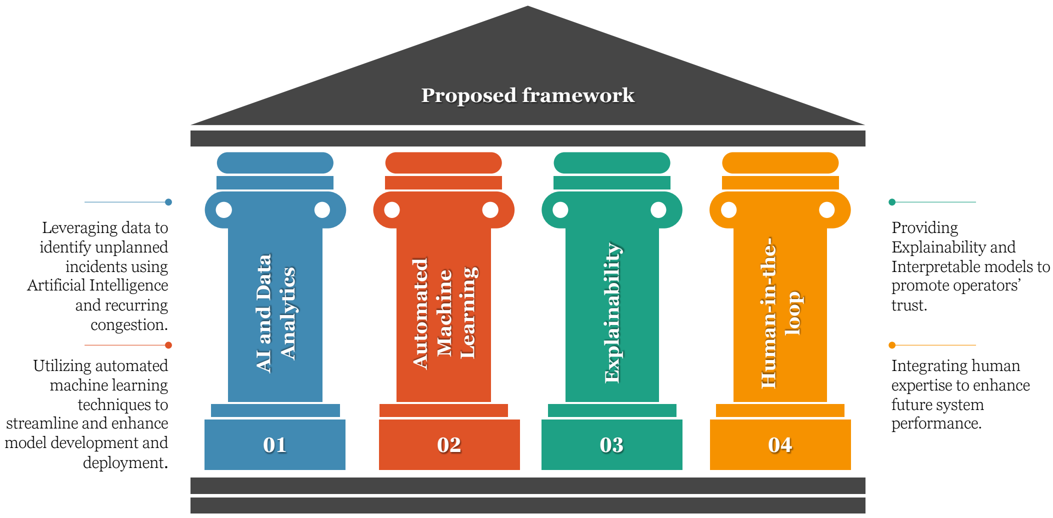



**Figure 4-1: The four pillars of our proposed framework.**

In the following sections, we provide a detailed presentation for each one of these pillars individually and their significance.

### *4.1.1 AI and Data Analytics*

AI and Data Analytics constitute the first pillar of the framework. Those are essential for the detection of traffic incidents, even before they occur. It involves the use of advanced data analytics and AI algorithms, which take as input large, even vast, quantities of historical and real-time traffic data. By finding patterns and trends, the system is capable of predicting potential incidents and providing real-time alerts to involved operators and affected users. Data Analytics focus on extracting meaningful insights from complex traffic data that include:

- Analysis of traffic data over time to identify patterns, trends, and abnormal behavior. In this context, time-series analysis is important to understand the daily and seasonal variation of traffic flows and to forecast peak congestion periods.
- Integrating spatial and temporal data is essential for comprehending the evolution and propagation of traffic incidents. This investigation aids for instance in identifying areas with a high frequency of incidents and presents the influence of spatial variables on traffic dynamics.
- Analyzing interrelations between factors related to traffic to identify factors which impact incidents. Such an understanding helps in building

more accurate predictive models and thus in implementing the measures for preventing and managing incidents in the context of strategic decision-making.

The analytics which aim to identify traffic incidents thus minimize the effects of traffic disruptions and increase the overall security of roads. The ability to foresee any accidents assists in controlling traffic, during the time of the incident taking place or even ahead of time, hence enabling authorities to apply preventive actions.

### *4.1.2 Automated Machine Learning*

The second pillar, Automated Machine Learning (AutoML), simplifies the process of developing and deploying machine learning models. AutoML automates the end-to-end process of applying machine learning to real-world problems, from data pre-processing and feature selection to model training and hyperparameter tuning. This automation reduces the time and expertise required to build effective prediction models, making ML accessible to a broader range of users. AutoML ensures that the best models from the pool that have been tried are used for incident detection, continuously improving their performance with minimal human intervention. This pillar is crucial for the efficiency and scalability of the incident detection system.

### *4.1.3 Explainability*

The third pillar focuses on making the AI models used in traffic management understandable to users. Techniques like SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-agnostic Explanations) are utilized to clarify how models make their predictions. SHAP provides information about the contribution of each feature, such as traffic volume, to the model's output. LIME, on the other hand, simplifies complex models to explain individual predictions, making it easier for any user to understand how specific factors influence the resulting prediction.

This transparency is essential for building trust and ensuring that the system's predictions are reliable. By understanding the rationale behind AI decisions, users can better validate the model’s recommendations and identify potential areas for improvement. Explainability also facilitates model evaluation and continuous refinement, leading to more accurate and reliable predictions in the context of any traffic management system.

### *4.1.4 Human-in-the-Loop*

The fourth pillar integrates Human-in-the-Loop (HITL) methodologies into the system. This pillar ensures that human expertise and feedback are incorporated into the prediction process, enhancing the system's accuracy and reliability. Key aspects include involving traffic management operators in reviewing and validating or adjusting, if needed, the model's predictions. Human input helps correct any errors in the model. This creates a feedback loop that enhances model performance over time, as operators provide validations and corrections. Lastly, combining HITL and explainability ensures that the incident detection system is not only highly effective but also trusted and accepted by its users.

The proposed framework, built on these four pillars, represents a comprehensive approach to automatic traffic incident detection. By integrating the key points mentioned as part of the four pillars, the framework addresses some of the major challenges and research gaps in traffic management, as stated in Section 2.4. It ensures that the system is proactive, efficient, accurate, and trustworthy, ultimately leading to improved road safety and better traffic management.

## *4.2 Proposed methodology*

The goal of our proposed approach is to enhance traffic management systems by leveraging advanced analytics and AI techniques, while ensuring the inclusion of explainability and human expertise in such automated systems. The proposed methodology in addition to the individual steps are illustrated in Figure 4-2. The process is complex and involves several phases, each critical to the system's overall effectiveness. In the following sections, we detail each phase.

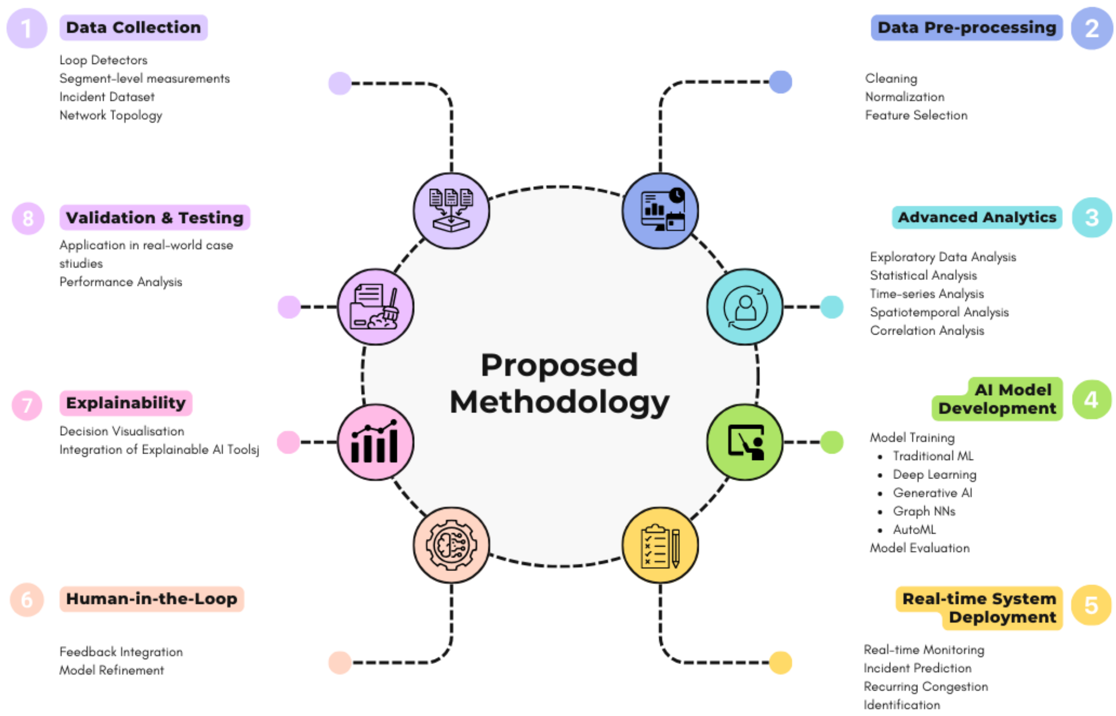



**Figure 4-2: Our proposed methodology.**

### *4.2.1 Data Collection*

Data collection is one of the paramount steps of any application which entails data-driven methodologies and Machine Learning. For the incident detection task, diverse and high-quality data sources need to be collected. The proposed approach is built upon the availability of the following datasets:

- **Inductive Loop Detector Measurements**: Loop detectors are devices embedded in road surfaces, and collect continuous data on vehicle count, speed, and occupancy. Loop detectors provide granular traffic data crucial for understanding real-time traffic dynamics and detecting anomalies indicative of incidents.

- **Segment Level Measurements**: Segment level data offers insights into specific road sections, allowing for more localized incident detection. These measurements might include average speed, vehicle density, and flow rates, which help identify sudden changes often associated with incidents.

- **Incident Dataset**: Historical incident data are essential for training data-driven AI-based models. This dataset includes detailed records of past incidents, such as accidents, road blockages, and breakdowns, including their locations, times, causes, impact and potentially resolutions. These data help the deployed models learn to recognize patterns and triggers of incidents.

- **Network Topology**: Information about the road network's structure, including the locations of the loop detectors, the layout of lanes, intersections, and connectivity, provides context for interpreting traffic data. Network topology helps in understanding how incidents in one part of the network can affect other parts.

### *4.2.2 Data Pre-processing*

After the data have been collected, the data pre-processing phase aims to transform raw data into a usable format for analysis and modeling. It includes several crucial steps:

- **Cleaning**: This step involves removing noise and correcting errors in the data. Techniques include handling missing values, removing duplicates, and correcting erroneous entries. As it is widely agreed, clean data are essential for ensuring accurate and reliable model training.

- **Normalization**: Standardizing data across different sources to a common scale ensures consistency.

- **Feature Extraction**: This process involves identifying and deriving relevant features from the raw data that will be used in model training. Features might include average speed, traffic density in adjacent detectors, weather conditions, and time of day, amongst many others. Effective feature extraction is crucial for enhancing model performance.

### 4.2.3 Advanced Analytics

Before proceeding with the development of AI models, it is crucial to understand the available data through advanced analytics, which include:

- **Exploratory Data Analysis (EDA)**: EDA is a standard step in every data-driven analysis and involves visualizing and performing basic descriptive analytics the data to uncover patterns, trends, and anomalies. Techniques include plotting histograms, scatter plots, and heatmaps. EDA helps in understanding the underlying distributions and relationships in the data.

- **Time-Series Analysis**: Given that traffic data is inherently temporal, time-series analysis is crucial. It involves analyzing data points collected or recorded at specific time intervals to identify trends, seasonal patterns, and cyclical behavior. Techniques such as ARIMA (AutoRegressive Integrated Moving Average) models are commonly used for this purpose.

- **Statistical Analysis**: Applying statistical methods to analyze data helps in understanding correlations and dependencies between variables. For example, correlation analysis can reveal how the time of day might influence traffic speed and incident rates.

- **Spatiotemporal Analysis:** Integrating both spatial and temporal dimensions to understand how traffic patterns evolve over time and across different locations. This analysis is crucial for identifying how incidents in one area might affect traffic flow in adjacent areas. Techniques involve the use of geospatial data and time-series data to create models that can predict traffic conditions based on both location and time. This is particularly useful for managing traffic during large events or in areas with frequent recurring congestion.

- **Correlation Analysis:** Examining the relationships between different variables to determine how changes in one variable may affect others. For instance, analyzing the correlation between traffic characteristics and incident occurrence can help identify potential predictors of traffic incidents.

Correlation analysis helps in understanding the interdependency between various factors influencing traffic and can be used to enhance the accuracy of incident prediction models.

### *4.2.4 AI Model Development*

Developing robust AI models for the timely and accurate detection of traffic incidents is positioned at the core of our proposed approach. This phase involves:

- **Algorithm Selection**: Choosing the right machine learning algorithms based on the problem and data characteristics is of utmost importance. Common algorithms for incident detection include decision trees, support vector machines, neural networks, and ensemble methods, as seen in the extensive literature review (Chapter 2).
- **Model Training**: Training involves feeding the pre-processed data into the selected algorithms to learn patterns and make predictions. Training requires splitting data into training, testing and validation sets to ensure the model can generalize to unseen data.
- **Parameter Tuning**: Adjusting the hyperparameters of each model to optimize its performance. Techniques such as grid search and random search are used to find the best combination of parameters.
- **Model Evaluation**: Assessing the model's performance using metrics like precision, recall, F1-score, and accuracy. This constitutes a critical step since it ensures the model is accurate and reliable in detecting incidents.

### *4.2.5 Real-time System Deployment*

Deploying the developed and trained models in a real-time environment involves several steps:

- **Real-time Monitoring**: Implementing systems to continuously monitor traffic conditions. Real-time data streams are processed to detect incidents as they occur. Moreover, establishing processes for regular updates and maintenance of the system is important to make sure that the system remains reliable. This includes updating models with new data, retraining as necessary, thus ensuring the system is operating smoothly and still produces reliable outputs.

- **Incident Prediction**: Automatically detecting and identifying incidents in real-time. The system should provide timely alerts and insights to traffic management operators for quick response and mitigation.

- **Recurring Congestion Identification**: Automatically detecting recurring congestion in real-time in any part of the transportation network. It involves analyzing traffic data to detect patterns and trends of congestion that occur regularly at specific times or locations. The proposed system can predict recurring congestion events, such as daily rush hours. Identifying these patterns allows for proactive measures to be implemented, such as adjusting traffic signals, providing real-time traffic alerts to drivers, and optimizing alternative routes.

### *4.2.6 Human-in-the-Loop*

Incorporating human expertise enhances the system's performance and reliability, in the following ways:

- **Feedback Integration**: Collecting and integrating feedback from traffic management professionals and other stakeholders to continuously improve the model. Human feedback helps in refining model predictions and reducing false positives/negatives.

- **Model Refinement**: Continuously updating and refining the model based on new data and feedback. This iterative process ensures that the model adapts to changing conditions and improves over time.

### *4.2.7 Explainability*

Ensuring the AI model's decisions are understandable and transparent is crucial for trust and accountability, in the following respect:

- **Decision Visualization**: Visualizing the model's decision-making process helps in making its predictions interpretable. Techniques include feature importance plots, and dependence plots.

- **Integration of Explainable AI Tools**: Using tools and techniques like SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-agnostic Explanations) to explain the model's predictions. These tools provide insights into which features are driving model decisions, helping stakeholders understand and trust the AI system.

### *4.2.8 Validation & Testing*

Finally, the proposed system should undergo rigorous validation and testing to ensure its effectiveness and robustness. This phase includes:

- **Application in Real-world Case Studies**: Testing the model in real-world scenarios to validate its performance. Case studies provide practical insights into how the model performs under various conditions and help identify areas for improvement.

- **Performance Analysis**: Analyzing the system's performance over time and across different conditions. This involves monitoring key performance indicators (KPIs) like detection accuracy, f1-score, and system reliability stemming from user feedback. Continuous performance monitoring helps in maintaining and, potentially, enhancing the system's effectiveness.

# 5 AI-Driven Traffic Incident Detection for Planned and Unplanned Events

In this Chapter, the proposed work of the Thesis concerning the application of Artificial Intelligence (Machine Learning and Deep Learning) techniques for the prompt identification of traffic incidents is presented. This chapter discusses the need of using AI to uncover patterns in traffic dynamics in addition to the advantages compared to traditional methods and presents the foundation of the proposed approach. An overview of the entire approach is provided, along with a detailed analysis of the detection of both planned and unplanned incidents. The developed approach has been tested and verified in real-life case studies in two urban environments, and the findings are discussed and displayed in Chapter 9, while the technical implementation details of the information system developed are presented in Chapter 8.

## *5.1 Introduction and Motivation*

The rapid urbanization and expansion of cities have led to increased traffic congestion and a higher occurrence of traffic-related incidents. Traditional traffic management systems often struggle to cope with the dynamic nature of urban traffic, especially when it comes to detecting and responding to traffic incidents in real time. This has created a pressing need for more advanced, efficient, and reliable methods of traffic incident detection.

The use of Artificial Intelligence (AI), particularly Machine Learning (ML) and Deep Learning (DL), has paved new ways for addressing these challenges. AI-driven methodologies can analyze vast amounts of traffic data, uncover hidden patterns, and provide timely and accurate detection of traffic incidents. This is crucial for mitigating the effects of traffic incidents, such as delays, economic losses, and safety hazards.

The motivation for this research stems from the recognition that Artificial Intelligence (AI), particularly Machine Learning (ML) and Deep Learning (DL), offers unprecedented capabilities for addressing the challenges faced by traditional traffic management systems, as illustrated in Section 2.4. More specifically, AI-driven methodologies excel at processing big amounts of data and uncovering hidden patterns that are not easily detectable through conventional means. These capabilities are crucial for the timely and accurate detection of traffic incidents.

One of the primary motivations for employing AI in traffic incident detection is the enhancement of detection capabilities. Unlike traditional methods that rely on static sensors and manual inputs, AI methodologies can analyze large datasets continuously and in real time. This allows for the identification of traffic patterns and anomalies that indicate incidents much faster and more accurately than human operators or conventional means. Another key motivation is the ability of AI systems to facilitate real-time monitoring and response. The quick processing of data and the ability to make data-driven decisions in real time can significantly reduce the response time to traffic incidents. This is essential for minimizing the impact of incidents on traffic flow and enhancing overall traffic management efficiency. AI systems can promptly and automatically alert traffic management centers about incidents, allowing for quicker deployment of emergency services and traffic rerouting measures. Moreover, these models are able to capture both spatial and time complexities which are inherent to the task of AID in traffic management.

The scalability and adaptability of AI-driven systems also provide a compelling motivation for their use. These systems can be scaled to handle varying volumes of data and can be adapted to different urban environments, making them suitable for deployment in cities of different sizes and with diverse traffic conditions. As more data is collected and analyzed, AI systems can continuously improve their accuracy and efficiency, further enhancing their utility in dynamic urban settings. Additionally, the integration of AI-driven traffic incident detection systems with broader smart city initiatives represents a significant motivation. By linking traffic management systems with other smart city technologies, cities can achieve a more cohesive and efficient urban management infrastructure. This integration can enhance not only traffic management but also public safety, environmental sustainability, and the overall quality of urban living.

This chapter aims to provide a comprehensive overview of the application of AI-driven methodologies in traffic incident detection, for both planned and unplanned events. The chapter will present an in-depth analysis of how specific AI models, widely employed in relevant studies and research papers, are applied to detect incidents. Special attention is given to distinguishing between the detection processes for planned incidents, such as recurring congestion and unplanned incidents, such as accidents or sudden road closures. The insights derived from this study aim to contribute significantly to the development of more effective, responsive, and intelligent traffic management systems which make use of trained AI models in the detection of incidents.

## 5.2 Data-driven Algorithms for unplanned non-recurring incident detection

Based on the extensive literature review we have conducted which is documented in Chapter 2, we have selected some of the most widely used state-of-the-art algorithms and methods to analyze and include in our proposed framework. For this reason, we have chosen not to focus on comparative or time-series algorithms, since although these have been used extensively in the past, there has been a shift in Machine Learning and Deep Learning approaches. Another reason we have taken the decision to not include the widely used California algorithms is the fact that these need as input only occupancy observations and do not take into account other traffic characteristics, which, in many scenarios are not reliable enough to base our analysis on. Instead, we have placed our focus on the following Machine Learning and Deep Learning algorithms, including Supervised (the widely employed SVM and a suite of Neural Networks) and Unsupervised approaches for anomaly detection, for instance Isolation Forest.

In this section, we present a high-level overview of the selected algorithms in addition to the rational on which we have based the selection.

#### 5.2.1.1 SVM

Support vector machine (SVM) is a supervised approach which is constructed from a unique learning algorithm that extracts training vectors that lie closest to the class boundary and makes use of them to construct a decision boundary that optimally separates the different classes of data.

Concerning how the SVM operates, consider the problem of incident detection where X is an input vector with n dimensions. The SVM performs the following operation involving a vector $W=\{w_1,\ldots,w_n\}$ and scalar b:

$$f(X)=\mathrm{sgn}(W\cdot X+b)$$

Positive sign of f(X) may be taken as incident state while negative value of f(X) may be regarded as incident-free.

We have chosen this algorithm, since, as illustrated from the literature review, results from various studies have shown that SVM offers a lower misclassification rate, higher correct detection rate, lower false alarm rate and slightly faster detection time than other models in traffic incident detection (Yuan & Cheu, 2003).

#### 5.2.1.2 Isolation Forest

In comparison to other anomaly detection methods such as Support Vector Machines and Decision Trees which require a labelled data set to form a classifier, Isolation Forests are generally used in an unsupervised manner. Isolation forests only require a few conditions to separate anomalies from normal observations when compared to other methods which use basic distance and density measures.

There are several works in the field of AID which use Isolation Forests. Their low linear time complexity and small memory requirements aid in eliminating major computational cost of distance calculation in all distance and density-based methods. Lastly, isolation forests are able to perform well in a multi-dimensional feature space.

#### 5.2.1.3 Neural Networks with Wavelet transformation

With capabilities of learning, self-adaptation, and fault tolerance, the Artificial Neural Networks (ANNs) approach has demonstrated good performance in many pattern classification applications, including several works in the field of traffic incident detection. In the study by (Ki, Jeong, Kwo, & Kim, 2019), a three-layered ANN model for incident detection was developed, as shown in Figure 5-1.

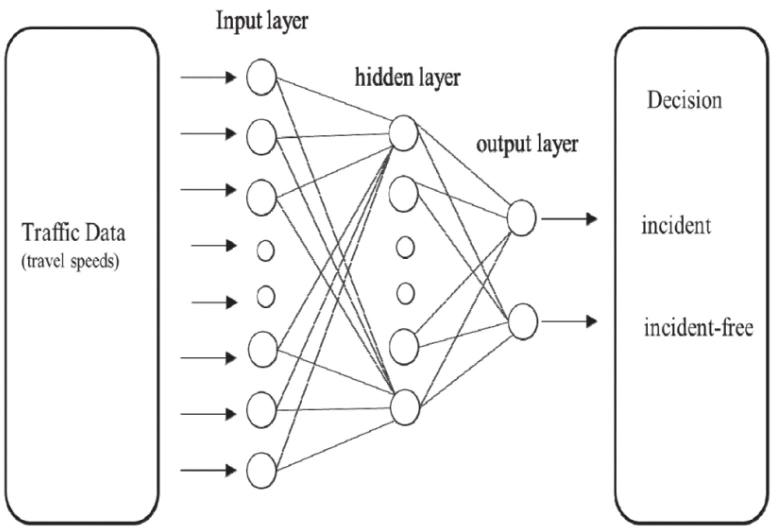



**Figure 5-1: Artificial Neural Network for Freeway Incident Detection (Ki, Jeong, Kwo, & Kim, 2019).**

Wavelet Neural Networks (WNN) are a class of networks that combine the classic sigmoid artificial neural networks (ANNs) and the Wavelet Analysis (WA). Wavelet analysis reveals the frequency components of signals just like the Fourier transform, but it also identifies where a certain frequency exists in the temporal or spatial domain. WNNs have been used with great success in a wide range of applications, since Wavelet Analysis has proved to be a valuable tool for analyzing a wide range of time-series and has already been used with success in image processing, signal denoising, density estimation, signal and image compression and time-scale decomposition. There is a correspondence between wavelet scales and frequency, such that a smaller scale corresponds to a compressed wavelet, which is high in frequency, while larger scales correspond to a stretched wavelet, representing lower frequency. Scales are often converted to spatial frequencies for better interpretability. By means of wavelet transformation, time series can be decomposed into a time dependent sum of frequency components. As a result, we are able to capture seasonalities with time-varying period and intensity.

In our case, we have used the Python library PyWavelets, which is open- source wavelet transform software for Python.

#### 5.2.1.4 BCNN

Another model which is frequently used in literature are the Convolutional Neural Networks (CNNs). According to Oquab et al. (2014), a CNN is an algorithm that excels in image processing, computer vision, and image recognition. Because CNN structures are sensitive to distance, they have primarily been applied to spatial problems. Convolution, pooling, and fully connected layers make up each CNN structure. Different features are extracted by different filters (also known as kernels) in the convolution layers. These filters are collections of learnable weights that are modified during training to generate output features. Prior to calculating the product between the numbers at the same location in the input matrix and the filter, the convolution filter is first positioned in the upper left corner of the input matrix. These products are then added together to produce a single number, which is the outcome of this operation's convolution. Then, the filter is moved to the right by one element, and the convolution result is obtained. A pooling layer is then used to extract dominant features and reduce the number of parameters. Then, the results are sent to the fully connected layer, which makes a prediction. Based on the type of pooling layer (average pooling or max pooling), the average or the maximum of the numbers at the same location in the output of the convolution layer matrix and the pooling kernel are calculated. The output dimensions of the pooling layer depend on the stride setting. Finally, the matrix is flattened and is passed on to the fully connected layer (Ansari Esfe, 2021). An example is illustrated in Figure 5-2.

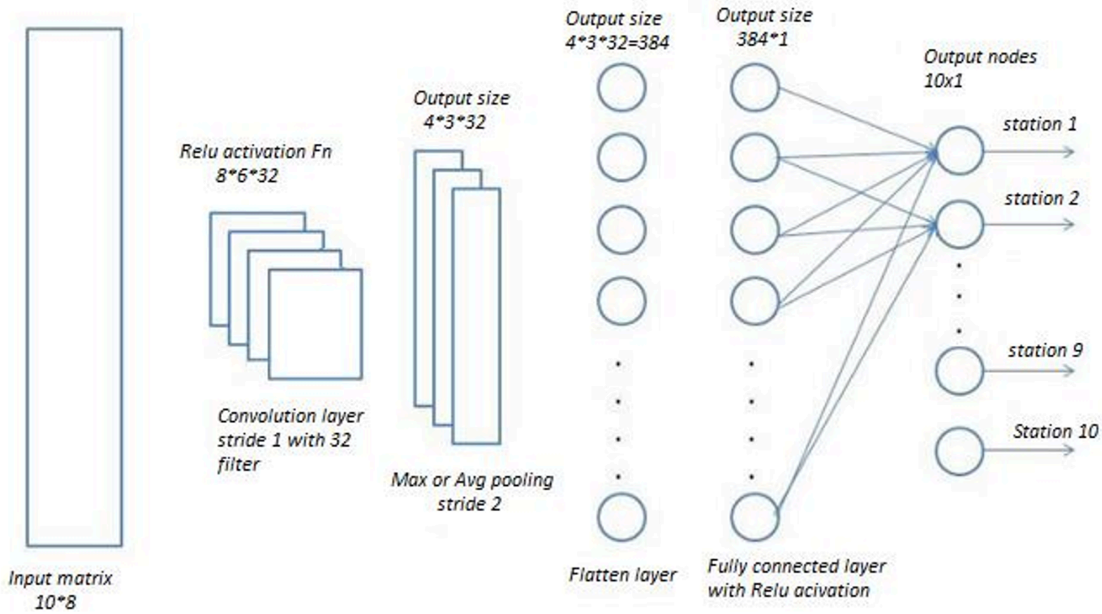



**Figure 5-2: Structure of a Convolutional Neural Network. (Ansari Esfe, 2021)**

In our case, we have chosen to include a Bayesian Convolutional Neural Network (BCNN)., which integrates probabilistic models and deep learning to consider uncertainties from both model and data. (Liu, Jin, Li, Hu, & Lia, 2022) Bayesian deep learning models exploit probabilistic layers that are trained using Bayesian inference to capture uncertainty over weights and activations. Because these probabilistic layers are designed as alternatives to their deterministic layers, Bayesian deep learning models create a straight-forward way to extend traditional deep learning models to endorse probabilistic deep learning.

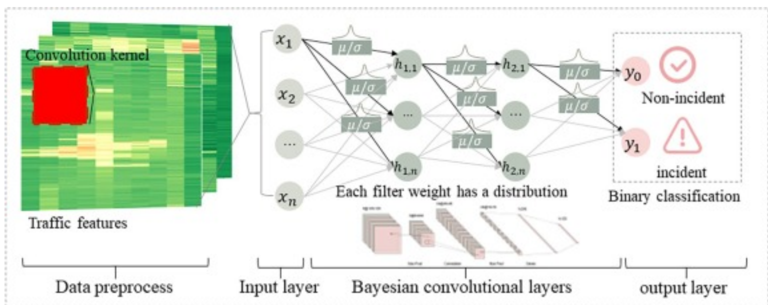



**Figure 5-3: Bayesian convolutional neural network model. (Liu, Jin, Li, Hu, & Lia, 2022)**

#### 5.2.1.5 Autoencoder

An Autoencoder is a generative deep learning algorithm used for reconstructing high-dimensional input data using a neural network with a narrow bottleneck layer in the middle which contains the latent representation of the input data.

It consists of an Encoder and a Decoder. The encoder network accepts high-dimensional input data and translates it to latent low-dimensional data. The input size to an Encoder network is larger than its output size. On the other hand, the Decoder network receives the input from the Encoder's output, which the Decoder's objective is to reconstruct the input data.

The Autoencoder accepts high-dimensional input data, compress it down to the latent-space representation in the bottleneck hidden layer; the Decoder takes the latent representation of the data as an input to reconstruct the original input data.

Therefore, Autoencoders have been used for Anomaly Detection tasks, by comparing the output from the Decoder and the input to the Network and using a threshold, either manually set or learnt from the data itself. If the loss value exceeds the threshold, then the instance is categorized or classified as an anomaly. In that sense, we can say that the Autoencoder works on an unsupervised manner, taking into account that it uses future values of the observations dataset, and the classification is based on a manually set threshold.

#### 5.2.1.6 Bidirectional LSTM

Fully connected neural networks (FCN) are a combination of many neurons in consecutive layers. The neurons are connected in a way that enables the model to solve complex, non-linear problems.  However, the FCN model structure cannot consider the hidden relationships among time steps in a time series of data. On the other hand, for such cases, it is important to take into account those relationships. A Recurring Neural Network (RNN) considers that the data in the sequence are related to each other. While RNN structures are appropriate for time series prediction, RNN can be defined as a very deep FCN with more time lags (hidden layers). Thus, RNN results in a vanishing or exploding gradient of the network, which means that the accuracy of a simple RNN may decrease as the sequence length increases because the earlier cells in the RNN get a small gradient update and stop learning in the backpropagation process. Thus, gated recurring neural network models, such as LSTM, have been proposed to overcome this issue. (Ansari Esfe, 2021) The internal structure of a LSTM network is shown in Figure 5-4.

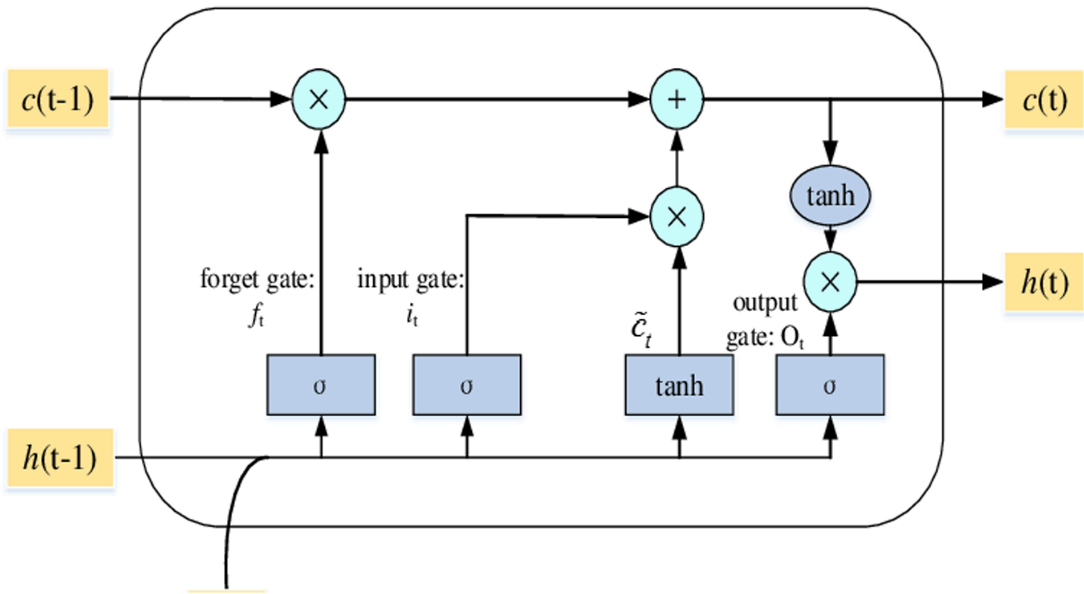



**Figure 5-4: Internal Structure of LSTM. (Yuan, Li, & Wang, 2019)**

Bidirectional LSTMs are based on the traditional LSTMs that were introduced to improve model performance on sequence classification problems (Huang, Xu, & Yu., 2015). The arrangement of the LSTM memory block enables the network to store and retrieve information over long periods (Figure 5-5). One drawback of the standard LSTM networks is that they only have access to the previous context but not to future context. On the other hand, Bidirectional LSTMs can capture both forward and backward dependencies in time series data (Cui, Ke, & Wang, 2018). In problems where all time steps of the input sequence are available, Bidirectional LSTMs train two instead of one LSTMs on the input sequence (i.e., the input sequence as-is and a reversed copy of the input sequence).

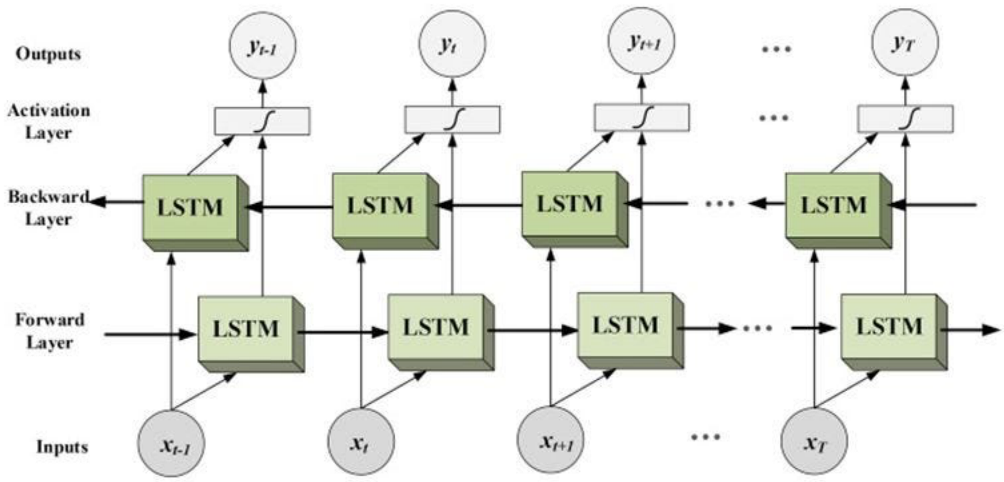



**Figure 5-5: The architecture of a Bidirectional LSTM model.**

The bidirectional LSTM is a good fit for traffic state prediction as it can potentially capture temporal autocorrelation in the data. Once the model is trained based on the historical data, the future values are estimated. Thereafter, the anomalous behavior can be classified by setting a threshold for loss values and examining the actual traffic data with the corresponding pattern.

#### 5.2.1.7 Graph Neural Networks

Graph Neural Networks (GNNs) have emerged as a powerful tool in enhancing incident detection in traffic systems. GNNs are particularly adept at handling data that is structured as graphs, which is a natural representation for traffic networks where nodes can represent detectors and edges can denote the roads connecting them. This structure allows GNNs to learn complex patterns of traffic flow and interactions between different parts of the network. By leveraging the spatial dependencies and the temporal dynamics of traffic data, GNNs can more accurately predict incidents, such as traffic jams or accidents, in real-time. This capability not only improves the efficiency of traffic management systems but also enhances road safety.

In graph-based modeling for traffic systems, each traffic sensor is represented as a node within the network graph, with the connections between roads depicted as edges linking these nodes. In Figure 5-6, a general pipeline of SpatioTemporal GNN models for traffic predictions are illustrated. A significant benefit of employing traffic sensor data in this context is the straightforward use of the collected traffic data as attributes for the nodes, avoiding the need for extensive preprocessing. However, there are caveats to consider, such as the fact that the placement of traffic sensors is restricted by various factors, including the cost of installation. (Jiang & Luo, 2022)

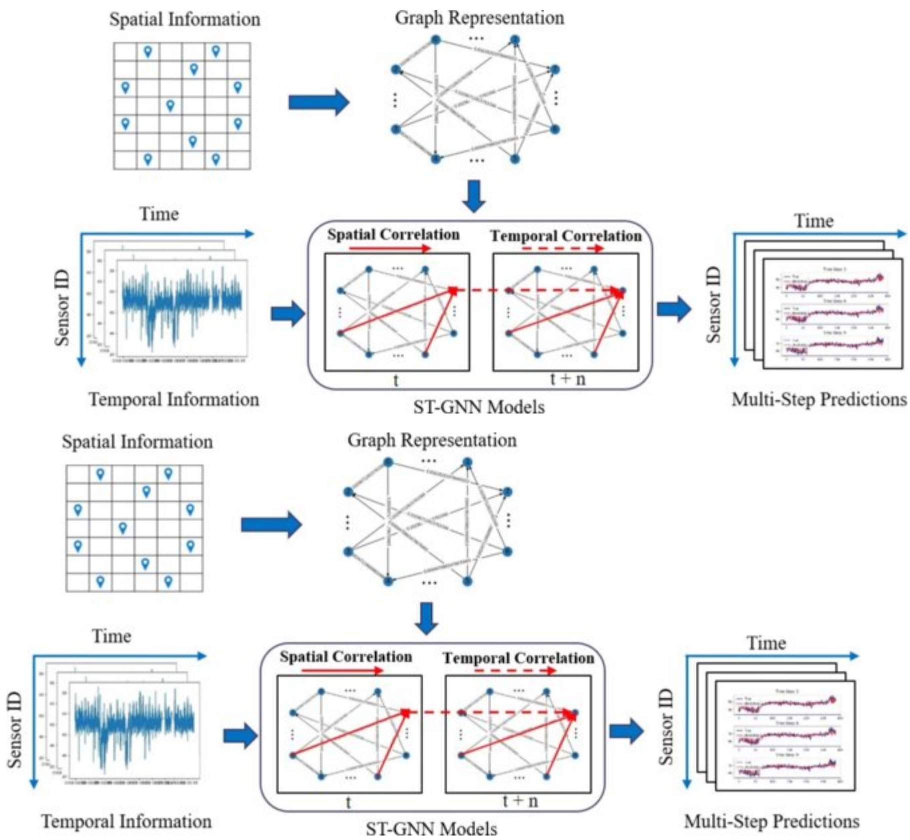



**Figure 5-6: A general pipeline of GNN models for traffic prediction. (Bui, Cho, & Yi, 2021)**

#### 5.2.1.8 AIMSUN algorithm

Aimsun is a leading company in the field of traffic management providing micro, meso and macro-simulations and respective analytics. Aimsun has developed advanced solutions for real-time traffic incident detection and management, notably through its Aimsun Live and Aimsun Predict platforms.

Aimsun Live is a real-time predictive traffic management solution that utilizes live and historical data to simulate and monitor traffic networks. It provides immediate forecasts of upcoming traffic conditions, enabling traffic management centers to proactively address potential issues before congestion arises. On the other hand, Aimsun Predict processes real-time data to forecast future traffic states, offering situational awareness and supporting proactive decision-making. It employs data cleaning, clustering, prediction, and incident detection techniques to interpret real-time data, providing alerts for unusual network performance and potential incidents. Aimsun Predict's online incident detection capabilities involve identifying sudden changes in traffic data—such as sharp drops in traffic flow outside peak hours—that may indicate incidents like crashes or collisions.

Existing incident detection methods within the Aimsun framework deployed state-of-the-art techniques based of the California #7 algorithm as well as probabilistic methodologies. Such techniques are well-known to have limitations both with unreliable data and in real-time scenarios. Within the scope of the FRONTIER project, Aimsun has been experimenting with a methodology that is able to cope with current state-of-the-art limitations in the area by being trained in an unsupervised way (i.e., independently of having labelled datasets), working independently on different timeseries (e.g., traffic flow, or speed, or occupancy) and working without the need of close sensor-pairs. Results of such experimentation are used to provide a baseline for further development and integration of incident detection algorithms in a real-time scenario.

Aimsun's baseline has been built upon the assumption that incident detection can be regarded as a transformation of individual timeseries to a space where the distance between structural outliers is magnified independently on each input variables. According to (Herrmann et al. 2022), two types of anomalies or outliers can be distinguished: distributional outliers and structural outliers. Distributional outliers look like normal data (i.e., inliers), but are in a low-density region in the data (or embedding) space. Since distributional outliers are structurally like inliers, their detection requires a lot of normal data for an accurate estimation of the probability density function and the setting of the probability threshold that defines the frontier of inliers.

On the other hand, structural outliers are those that belong to a manifold[7] different from the one formed by inliers. Road traffic incidents belong to the latter category, whose detection is not equally easy for all traffic variables. For example, road traffic occupancy usually offers easier incident detection than traffic flow. Similarly, occupancy difference between sensor-pairs offers an easier detection than single-sensor occupancy. Moreover, different embedding spaces can offer distinct levels of detection power. (Torrent-Fontbona F. , Dominguez, Fernandez, & Casas, 2022)

## *5.3 Advanced Analytics Methods for Recurring Congestion Identification*

Recurring congestion is a common issue in the transportation sector, particularly in urban areas with high traffic volumes. This type of congestion typically arises due to routine traffic demand patterns, such as morning and evening rush hours. Unlike non-recurrent congestion, which is caused by unpredictable events like accidents or weather disturbances, recurrent congestion occurs regularly and predictably. The presence of recurrent congestion not only affects the efficiency of the transportation network but also leads to increased fuel consumption, higher emissions, and longer travel times for commuters.

Several systems have been developed to identify and monitor recurrent congestion, some of which employ Machine Learning and Deep Learning techniques. These studies primarily concentrated on identifying the dominant trend in congestion spread. For example, (Liu, Zheng, Chawla, Yuan, & Xie, 2011) created an algorithm to identify a causal pattern for traffic situations. They segmented the city of Beijing into multiple areas and charted these areas. In the diagram, nodes symbolized the areas, and edges illustrated the movement of traffic among the areas. They suggested utilizing the spatiotemporal outlier tree (STOTree) along with a frequent subtree

[7] A manifold is a topological space that is modelled closely on Euclidean space locally but may vary widely in global properties.

algorithm to uncover the causal anomaly pattern in road systems. Additionally, (Nguyen, Liu, & Chen, 2016) presented a method to identify congested roads over time and the causal links between them. They created the spatiotemporal congestion (STC) algorithm, which produced the most common sub-structures (subtrees) from all detectable tree structures in a network to uncover the repeating propagation pattern. Moreover, they employed a dynamic Bayesian network (DBN) to assess the likelihood of each propagation happening. By merging these two methods (STC_DBN), they were able to effectively identify the congestion propagation pattern (Liang, Jiang, & Zheng, 2017) introduced a data-driven method that identified the cascading patterns of traffic flow by optimizing the likelihood function based on the available data. They asserted that this model surpassed the STC_DBN algorithm regarding accuracy and computation time. Subsequently, (He, Wang, Fang, & Li, 2018)enhanced the STC_DBN algorithms. They suggested a spatiotemporal congestion co-location pattern (STCCP) to identify the congestion propagation pattern. They created three-dimensional models incorporating the factors of time, place, and traffic. By utilizing the congestion characteristics found in neighboring areas and later periods, they analyzed the congestion pattern. Although the aforementioned studies primarily concentrated on identifying the common congestion patterns at the network level with a tree-based algorithm, they failed to anticipate congestion propagation.

Moreover, (Wang & Zhou, 2017) and (Ji, Wang, Zhou, & Chen, 2019) applied a mining algorithm to identify the spatiotemporal congestion. Additionally, they established speed characteristics using taxi trajectory data to identify the spatiotemporal congested regions for constructing the frequent patterns. By integrating the common congestion patterns with the rules of congestion propagation, researchers can foresee congestion spread during repeated events. Although recent research has concentrated on creating frequent trees or identifying the most likely congestion propagation patterns to forecast congestion issues, (Xiong, Vahedian, Zhou, Li, & Luo, 2018) suggested an effective algorithm to anticipate where congestion will spread in the near term. They introduced the idea of a propagation graph (Pro-Graph) to represent the direction of congestion propagation in networks. At each time period, they forecasted every Pro-Graph that could be identified in the network through empirical probabilities of dissemination computed from past data. The latter is among the groundbreaking studies in predicting congestion propagation. Nonetheless, like earlier studies, they did not assess this model for non-recurring events. Recently, (Majumdar, Subhani, Roullier, Anjum, & Zhu, 2021) employed LSTM networks with univariate data (speed) and multivariate data (speed and flow) to forecast congestion spread across road networks. Initially, they forecasted vehicle speed for two sensor locations. Next, they examined the speed patterns to show the similarity in the speed profiles of the two locations. The time intervals between comparable anomalies, like a decrease or rise in velocity, were determined. Time delays were subsequently utilized to evaluate the congestion spread duration. Although this study forecasts the spreading of congestion, this model is not applicable to arterial roads due to the presence of signalized intersections. Furthermore, while this research simulates congestion spread at the network scale, it does not pertain to nonrecurring incidents.

Nonetheless, a significant amount of research focuses on simpler and more basic data-driven techniques, some of which are employed in our analysis and are outlined below. Descriptive analytics methods are essential for understanding the current state and historical trends of traffic congestion. It involves summarizing and visualizing data to identify patterns and insights, forming the basis for more advanced analytical methods. One of the simplest yet effective approaches constitute of Exploratory Data Analytics and more specifically the visualization of traffic over a specific period of time. By plotting traffic occupancy, speed or flow over time, one can visually identify patterns of congestion. This could be a time-series plot showing traffic volumes every day at different times. Regular spikes at specific times might indicate recurrent congestion due to rush hours. Another effective way to visually analyse recurring congestion is using histograms and distribution plots; histograms can provide insights into the distribution of traffic volumes or speeds. Furthermore, heatmaps can be especially useful for visualizing traffic patterns across days and times, providing a clear and intuitive way to identify congestion hotspots. For example, a heatmap with days of the week on one axis and times of the day on the other can quickly show when congestion is most likely to occur. By plotting box plots for traffic volumes or speeds for different times of the day or days of the week, one can identify variability and potential outliers. Periods with lower median speeds and high variability might be indicative of congestion. Investigating correlations between different variables might provide insights into factors influencing congestion. For instance, there might be a strong negative correlation between vehicle speed and vehicle count, suggesting that as the number of vehicles increases, the average speed decreases, leading to congestion. More fine-grained analysis can be made to deduce daily or weekly patterns and identify congestion. By plotting two variables against each other, like traffic volume and speed, one can visually identify patterns or relationships as part of the depicted scatter plot. A downward trend in such a plot might indicate that as traffic volume increases, speeds decrease, signalling congestion.

In the context of transportation, Time-Series analysis is particularly useful for understanding traffic flow variations over days, weeks, or even longer periods, being a statistical technique that deals with time-ordered data points. Thus, it can be used to reveal seasonal variations in traffic congestion, such as increased traffic during holiday seasons or reduced congestion during summer vacations, in addition to capturing daily peak hours. The advantage of time-series analysis is its simplicity and direct applicability to loop detector data, which is, by its nature, sequential. By employing moving averages, seasonal decomposition, or autocorrelation functions, one can identify periodic congestion patterns, trends, and seasonality. For example, a recurrent spike in traffic every weekday morning might indicate a congestion pattern due to work-related commutes. The STL decomposition breaks down a time series into three main components: trend, seasonal, and residual. The trend component shows the underlying trend in the data, abstracting away from the day-to-day or hour-to-hour fluctuations. If there's an increasing or decreasing trend over time, it will be observed as a steadily rising or falling line. The seasonal component captures the repeating patterns in the data and analyzing daily patterns could manifest as consistent peaks (e.g., during rush hours) and troughs (e.g., during the night) each day. What remains after the trend and seasonal components have been subtracted from the original data is the residual component.

A more advanced technique from the field of descriptive analytics to understand traffic patterns, is clustering. Clustering is a ML method which aims to group similar traffic patterns together, based on similarity measures, making it easier to identify common causes of congestion. For example, clusters might reveal that certain intersections consistently experience high traffic volumes during specific times of the day. Specifically, for our loop detector data we have employed k-means clustering. This algorithm partitions data into 'k' number of clusters. By segmenting traffic data into clusters, one might identify groups representing peak traffic times or nighttime inactivity. The elbow method is used to identify the optimal number of clusters. Each cluster can offer insights into specific traffic patterns, helping in congestion detection and management.

On the other hand, predictive analytics uses historical data to forecast future traffic conditions, allowing for proactive congestion management. It involves various statistical and machine learning methods to predict traffic flow, travel times, and potential congestion points. For instance, regression analysis models the relationship between traffic variables, such as volume, speed, and travel time. Linear regression is useful for direct relationships, while non-linear models can capture more complex interactions. In addition to regression analysis, Machine Learning algorithms are effective for predicting congestion by considering multiple variables and their interactions, capable of handling large datasets and provide insights into which factors are most influential in causing or propagating congestion. They can capture long-term dependencies and trends, making them highly effective for congestion forecasting. Those include among others SVMs, Decision Trees, Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs).

Moreover, as part of the analysis, to verify that we have managed to capture the time patterns correctly one can predict the traffic flow using conventional models, such as AutoRegressive Integrated Moving Average (ARIMA) models, widely used for time-series forecasting. They account for past values and trends to predict future traffic conditions. In order to perform the ARIMA forecasts, we plot the ACF (Autocorrelation Function), which shows how the values of the time series relate to their past values, and the PACF (Partial Autocorrelation Function), that illustrates the correlation between the series and its lags after accounting for the contributions from the intermediate lags. Moreover, the Augmented Dickey-Fuller (ADF) test needs to be performed.

While descriptive and predictive analytics techniques, which have been mentioned above, provide a comprehensive understanding of traffic patterns and future conditions, prescriptive analytics goes a step further by suggesting actionable interventions to mitigate congestion. Techniques in prescriptive analytics include optimization models for traffic signal timings, route optimization algorithms, and dynamic congestion pricing strategies. These methods are crucial for implementing effective congestion management solutions based on insights gained from descriptive and predictive analyses, however they are out of scope of our analysis and could be employed as future research.

To conclude, identifying and addressing recurring congestion is vital for improving urban mobility and reducing traffic-related issues. Descriptive analytics techniques, such as time-series analysis, heat maps, and clustering, provide a foundational understanding of congestion patterns. Predictive analytics methods, including regression analysis, machine learning algorithms, and time-series forecasting, enable accurate forecasting of future traffic conditions, thus enabling the identification of recurrent situations such as congestion during peak hours. Together, these techniques offer a robust framework for understanding and identifying recurring congestion.

# 6 AutoML-Driven Incident Detection

In the current Chapter, the proposed work of the Thesis concerning the application of AutoML techniques for incident detection is presented. This chapter addresses the rationale behind using AutoML, reviews relevant work, and discusses the theoretical foundation of the proposed approach. An overview of the entire methodology is provided, along with a detailed analysis of each component involved. The implementation process of the AutoML-driven incident detection system is thoroughly discussed. The developed approach has been tested and verified in real-life case studies in two urban environments, and the findings are discussed and displayed in Chapter 9.

## *6.1 Introduction and Motivation*

Efficient traffic incident detection in an automatic and prompt manner is paramount in urban traffic management, directly impacting congestion control and road safety. A traffic incident typically refers to any unexpected event that decreases road capacity and leads to congestion. Such incidents disrupt the flow of traffic, impede operations, and are responsible for not only delays but also increased pollution. Consequently, Intelligent Transport Systems (ITS) are increasingly focusing on reducing the impact of such traffic events. Thanks to the surge in available traffic data, Machine Learning has emerged as a powerful tool to improve upon the traditional algorithmic approaches, such as the California #7 Series (Balke, 1993), for detecting these incidents. However, the variable nature of traffic flow makes immediate and precise incident detection challenging. The widespread deployment of traffic sensors on highways has yielded extensive traffic flow data. Common data sources for identifying traffic incidents include stationary detectors like inductive loop detectors, GPS devices, and automatic identification systems such as Radio-Frequency Identification (RFID). Recent advances in machine learning have led to its accelerated adoption in the field of transportation engineering, with popular techniques encompassing Artificial Neural Networks (ANNs), Support Vector Machines (SVMs), Isolation Forests (IFs), and their various adaptations. While the traditional machine learning techniques have demonstrated utility in interpreting traffic flow data, they often require extensive domain knowledge for feature selection and model tuning, which can be a significant limitation given the variable nature of traffic patterns.

Automated Machine Learning (AutoML) is an emerging area in ML that seeks to automate the ML workflow from data preprocessing to model validation (Hutter, Kotthoff, & Vanschoren, 2019), thus enhancing performance and reducing the necessity for constant redesign. AutoML not only automates the meticulous process of discovering and fine-tuning the best-suited machine learning framework for the task at hand but also adapts as the characteristics of data evolve over time. Although setting up AutoML systems may initially demand more computational resources, the trade-off includes a substantial decrease in manual labor and the level of expertise traditionally required to develop high-performing models. Therefore, such automation provides robust AutoML methods that enable people, with either little or no specialized ML knowledge, to integrate ML solutions into data-driven processes. The latter is known as the democratization of ML (Hutter, Kotthoff, & Vanschoren, 2019) and it is aligned with the actual purpose of Artificial Intelligence: to learn and act automatically without human intervention (Song, Triguero, & Ozcan, 2019).

Despite the growing interest in AutoML in many fields, including transportation, a notable research gap exists in its application to traffic incident detection. To our knowledge, there are no comprehensive studies that have specifically tackled the use of AutoML for this purpose. While previous research has demonstrated the potential of AutoML in various domains such as healthcare, finance, and manufacturing ( (Hutter, Kotthoff, & Vanschoren, 2019), (He, Zhao, & Chu, 2021), (Karmaker Santu, et al., 2022)) , its application in the field of traffic management, particularly for incident detection, remains underexplored. This lack of research in applying AutoML to traffic incident detection presents a unique opportunity. The challenge lies in not only adapting AutoML to effectively analyze traffic data but also in validating its applicability across different urban settings. The conducted work aims to fill this gap by proposing a novel AutoML-based methodology in detecting traffic incidents. Moreover, the effectiveness of our proposed approach has been validated in two major European cities, in Athens, Greece, and Antwerp, Belgium, and the results are presented in Chapter 9; however, the methodology proposed is generic and could potentially be replicated or adapted for other urban environments. In doing so, this research contributes to the broader understanding of how AutoML can be utilized in urban traffic systems, potentially leading to more responsive and efficient traffic management solutions.

As far as I am aware, at the point of writing, this research represents a novel study aiming to address the automatic incident detection task by employing AutoML methodologies. Specifically, the main contributions of the work conducted can be summarized in the following:

1. **Integration of Data Pre-processing Techniques**: Recognizing the importance of data quality, we propose a data pre-processing algorithm before employing the AutoML process using TPOT (Tree-based Pipeline Optimization Tool). This integration aims to streamline the model development process, from raw data handling to final prediction.
2. **Contrast of our AutoML-based approach with General Approach Algorithms**: This research also sets to compare and contrast the performance and efficiency of AutoML frameworks against general machine learning algorithms. This analysis will help to elucidate the benefits and limitations of AutoML in the specific context of traffic incident detection.
3. **Assessment and comparison of our AutoML methodology across different urban contexts:** Last but not least, this work aims to explore the differences of the proposed automatic approach across two big European cities, Athens and Antwerp, which are presented as part of Chapter 9.

## 6.2 State-of-the-art Analysis

### 6.2.1 Theoretical Background on AutoML

AutoML has emerged as a transformative approach in the field of machine learning, aiming to automate the process of model selection and hyperparameter tuning. This section will delve into the general principles and methodologies of AutoML, highlighting its impact on accelerating and simplifying the deployment of machine learning models. AutoML is designed to make the machine learning (ML) process more accessible and automated, enabling experts in specific domains to leverage ML technologies without requiring extensive knowledge or a data analyst's assistance (Hutter, Kotthoff, & Vanschoren, 2019). At the heart of AutoML lies the challenge of Hyper Parameter Optimization (HPO), which involves the automatic tuning of hyperparameters to enhance the performance of ML systems across tasks like classification, regression, and time series forecasting (Hutter, Kotthoff, & Vanschoren, 2019). Recent advancements in AutoML have expanded its scope to include additional functionalities such as *Data Preparation, Feature Engineering, Model Generation* and *Model Evaluation.* (Hutter, Kotthoff, & Vanschoren, 2019) (He, Zhao, & Chu, 2021)

The primary goal of AutoML is to reduce the manual effort involved with machine learning technologies, thus accelerating their deployment. Consequently, various systems have attempted to minimize the work required to perform certain steps of the machine learning development workflow. For example, DeepDive/Snorkel (Ratner, et al., 2020) is a general, high-level workflow support system that helps users label and manage training data and provides high-level support for model selection. As previously mentioned, however, developing ML solutions still involves a lot of manual work. To design a truly automated system, it is important to address the bottlenecks in the current process. To better visualize these bottlenecks, we present, in Figure 6-1,a flowchart showing the end-to-end machine learning process. For each step in the flow, we outline the role of domain experts, the amount of manual work performed by the data scientist, and the communication required between the two. (Karmaker Santu, et al., 2022)

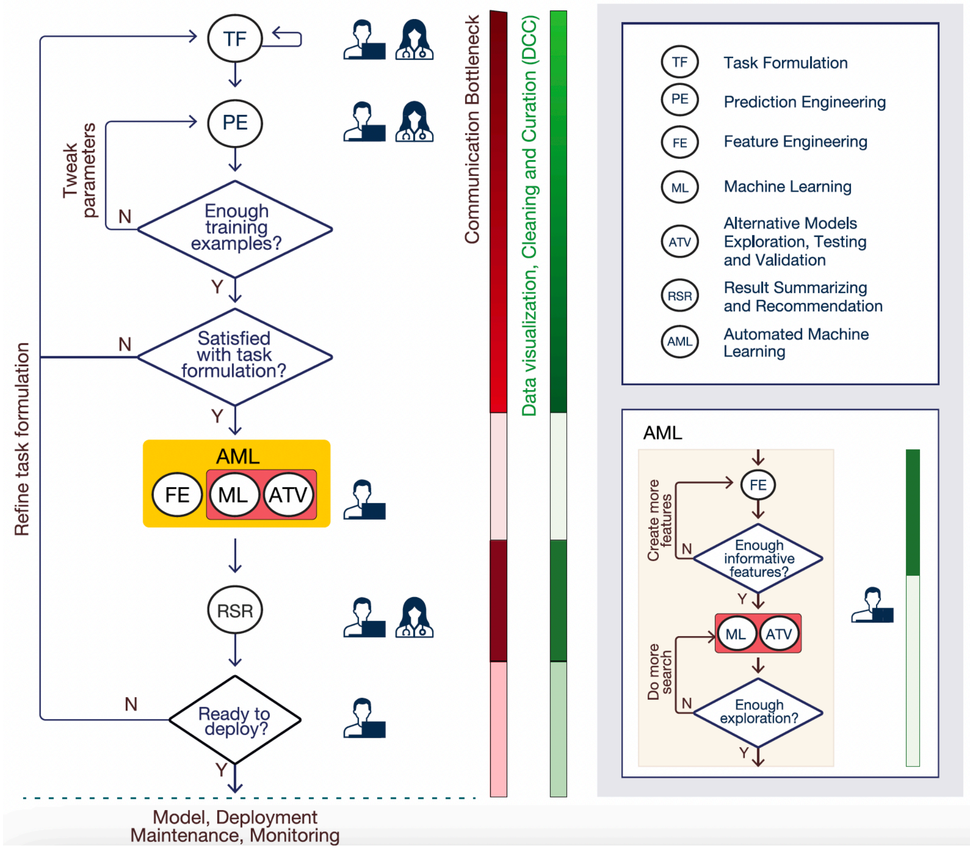



**Figure 6-1: Flowchart depicting the Machine Learning process, while highlighting points of interaction between domain experts and data scientists, along with bottlenecks. (Karmaker Santu, et al., 2022)**

The Data Preparation and Feature Engineering steps are associated with the available data used for the ML algorithms. The former includes actions for collecting, cleaning and augmenting the data, with the latter includes actions for extracting, selecting and constructing features. In the Model Generation step, a search is executed with the goal of finding the best performing model for the predictions, such as k-nearest neighbors (KNN) (Altman, 1992), Support Vector Machines (SVM) (Cortes & Vapnik, 1995), Neural Networks (NN), etc. The Model Evaluation step is responsible for evaluating the generated models based on predefined metrics and runs in parallel to the Model Generation step. The evaluation of the generated models is used for optimization of existing models and the construction of new models. The search procedure of AutoML terminates based on predefined restrictions, such as the performance of the models or the time passed.

As described above, AutoML deals with Model Selection Problem (MSP) as an optimization problem whose objective consists of finding the ML algorithm, from a pre-defined base of algorithms, and its hyper-parameter configuration that maximizes an accuracy measure on a given ML problem. In this sense, AutoML aims to improve the current way of building ML applications by automating the application of ML algorithms to datasets, in such a way that enables human users avoiding tedious tasks (e.g., hyper-parameter optimization). Although current AutoML methods have already produced impressive results, the field is still far from being mature. Regarding AutoML tools, the first AutoML method in tackling simultaneously the selection of algorithm and hyper-parameters was Auto-WEKA (Thornton, Hutter, Hoos, & Leyton-Brown, 2013).It uses Bayesian optimization to search for the best pair (algorithm, hyper-parameter setting), considering a base of 39 algorithms implemented in WEKA (a well-known open-source ML software that contains algorithms for data analysis and predictive modelling). Subsequently, Komer et al. (Komer, Bergstra, & Eliasmith, 2014) and Feurer et al. (Feurer, et al., 2015) developed Hyperopt-sklearn and Auto-sklearn, respectively. These two frameworks automatically select ML algorithms and hyper-parameter values from scikit-learn. In the case of (Komer, Bergstra, & Eliasmith, 2014), the AutoML method uses Hyperopt Python library for the optimization process, concretely a Bayesian optimization method as Auto-WEKA. Meanwhile, Auto-sklearn stores the best combination of ML algorithm and hyper-parameters that have been found for each previous ML problem, and using meta-learning it chooses a starting point for a sequential optimization process.

More recently, Sparks et al. (Sparks, et al., 2015) proposed a method that supports distributed computing for AutoML, and Sabharwal et al. (Sabharwal, Samulowitz, & Tesauro, 2016) developed a cost-sensitive training data allocation method that assesses a pair (algorithm, hyper-parameters setting) on a small random sample of the data-set, and gradually expands it over time to re-evaluate it when one combination is promising. Then, Olson and Moore (Olson, Bartley, Urbanowicz, & Moore, 2016) designed a framework for building and tuning classification and regression ML pipelines. It uses genetic programming to construct flexible pipelines and to select an algorithm in each pipeline stage. However, TPOT does not exhaustively test all different combinations of hyper-parameters which in turn causes that some promising configuration may be ignored.

Lately, Swearingen et al. (Swearingen, et al., 2017) built ATM, which is a collaborative service to build optimized ML pipelines. This framework has a strong emphasis on parallelization enabling the distribution of a single combination (algorithm, hyper-parameter setting) in a cluster to process it in a more efficient way. Currently, ATM uses the same base of algorithms from scikit-learn, and it finishes the optimization process after either a fixed number of iterations or after expending a time budget defined by the human user. One year later, Mohr et al. (Mohr, Wever, & Hüllermeier, 2018) developed ML-Plan, a framework for building ML pipelines based on hierarchical task networks. ML-Plan is initialized with a fixed set of pre-processing algorithms, classification algorithms, and their respective potential hyper-parameters. Nevertheless, ML-Plan only considers a supervised classification approach, ignoring the supervised regression perspective that, as it was stated before, is the most common approach in TF. From a technical perspective, AutoML attracted a lot of research interest resulting in several AutoML frameworks, such as: Autokeras (Jin, Song, & Hu, 2019), FEDOT (Nikitin, 2022) and TPOT (Olson, Bartley, Urbanowicz, & Moore, 2016). Additionally, research focusing on benchmarking several AutoML frameworks (Gijsbers, et al., 2019) (Zöller & Huber, 2021) concludes that they do not outperform humans yet but give promising results. (Fikardos, et al., 2022)

### *6.2.2 AutoML in Transportation and Traffic prediction studies*

In the transportation sector, AutoML's application is still on the rise. This section reviews the current state of research on the use of AutoML for traffic prediction and its role within the transportation domain. It will highlight studies where AutoML has been employed to optimize traffic flow, predict congestion, and improve overall transportation efficiency.

In the transportation area, to the best of my knowledge, only four papers have used AutoML methods for traffic forecasting (TF) ( (Angarita-Zapata, Masegosa, & Triguero, Evaluating automated machine learning on supervised regression traffic forecasting problems., 2020), (Angarita-Zapata, Triguero, & Masegosa, 2018), (Vlahogianni, 2015), (Angarita-Zapata, Masegosa, & Triguero, 2020)) .The first research carried out by Vlahogianni et al. (Vlahogianni, 2015) proposed a meta-modelling technique that, based on surrogate modelling and a genetic algorithm with an island model, optimizes both the algorithm selection and the hyper-parameter setting. The AutoML task is performed from an algorithms base of three ML methods (Neural Network, Support Vector Machine and Radial Base Function) that forecast average speed in a time horizon of 5 min, using a regression approach. After that, Angarita et al. in (Angarita-Zapata, Masegosa, & Triguero, 2020) and (Angarita-Zapata, Triguero, & Masegosa, 2018) used Auto-WEKA, an AutoML method that applies sequential model-based Bayesian optimization (Hutter, Hoos, & Leyton-Brown, 2011) to find optimal ML pipelines. Both papers compared the performance of Auto-WEKA w.r.t. the general approach, which consists of selecting by trial and error the best of a set of algorithms to predict traffic. In the case of (Angarita-Zapata, Triguero, & Masegosa, 2018) , the paper was centered in forecasting traffic level of service (LoS) at a fixed freeway location through multiple time horizons. On the other hand, in (Angarita-Zapata, Masegosa, & Triguero, 2020), the authors were focused on predicting traffic speed on a subset of families of TF regression problems focused on making predictions at the point and the road segment levels within the freeway and urban environments. Lastly, in (Angarita-Zapata, Masegosa, & Triguero, 2020), the authors focus on assessing Auto-sklearn's capability to recommend effective machine learning pipelines for traffic forecasting. This task is framed as a time series (TF) multi-class imbalanced classification problem, examined over various time horizons, spatial scales (both point and road segment), and in different environments (freeway and urban). The study tests three scenarios and findings indicate that Auto-sklearn's meta-learning component underperforms in handling TF problems, and optimization does not significantly enhance prediction accuracy.

All in all, it is certain that ML algorithms have played a crucial role in developing accurate models for automatic incident detection. However, some challenges persist, such as high computational costs and redundant model information, while minimizing human intervention. In response to these issues, adopting AutoML algorithms, which embody a pipeline model that automatically fine-tunes hyperparameters, presents a promising solution. Under this light, the conducted aims to contribute to the fast-growing field of AutoML by:

I. Developing AutoML-based prediction algorithms for the incident detection task, both from a regression and a classification standpoint;
II. Conducting of a comparison study between the proposed prediction methodology for each use case with other baseline methods; and finally,
III. Analyzing and assessing the models proposed in this research between different urban contexts, as presented in Chapter 9.

Last but not least, to the best of our knowledge, this study is the first one aiming to tackle the challenging problem of automatic incident detection using AutoML techniques.

## 6.3 Proposed Methodology

This work involves developing a methodology for automatic incident detection with the goal of identifying unplanned non-recurring events promptly and thus enabling a safer and more reliable Intelligent Transport Management system. The methodology flowchart, depicted in Figure 6-2, illustrates the general workflow of the present study. Initially, the process commences with data ingestion, followed by a thorough data-preprocessing stage to make the dataset suitable for model deployment. Subsequently, an AutoML framework, TPOT, is used as the foundation of our approach for model development and selection.

During the prediction phase, the data is divided into two sets: one for testing and the other for validation with unseen data. TPOT is deployed for crafting an effective machine learning model leveraging the training data, focusing on the problem from a regression and a classification perspective. After training and evaluating both the TPOT Classifier and TPOT Regressor, we compare their performance metrics to determine the most effective model for deployment. However, the benefits of utilizing TPOT extend beyond this initial selection phase.

AutoML systems like TPOT offer dynamic model updating, which can be crucial as traffic data evolves over time. This allows for continuous retraining and model refinement without the need to start the process anew, ensuring that the incident detection system can adapt to new patterns or changes in traffic flow. Furthermore, the ongoing use of AutoML provides an operational advantage in terms of parameter fine-tuning. TPOT, with its genetic programming-based optimization, can iteratively explore the parameter space to fine-tune the model as more data becomes available or as traffic conditions change, a process that is more resource-efficient and potentially more effective than manual tuning efforts. Lastly, the implementation of AutoML for ongoing model management allows for the incorporation of online learning techniques, where the model can be updated in real time with new data. This is particularly relevant for incident detection, where the timeliness of model updates can significantly impact the system's performance and reliability. Therefore, while the model selected via TPOT may be fixed during a specific period, our methodology is designed to facilitate model evolution, allowing for ongoing improvements and the incorporation of new data, which is a significant benefit over a static algorithmic approach.

It is important to highlight that in this work, the data-preprocessing step —which includes feature extraction, data sampling, and balancing—is carried out before training the models. This pre-processing is integral to both models' capability; however, a decision has been taken for pre-processing to be executed independently to guarantee compatibility, ensure uniformity across models, and ultimately, amplify their efficiency. A detailed diagram, as shown in Figure 6-3, provides an in-depth view of the modeling phase, illustrating the intricate steps involved in the training of both the classification and regression models, thus highlighting the dual approach of tackling automatic incident detection task.

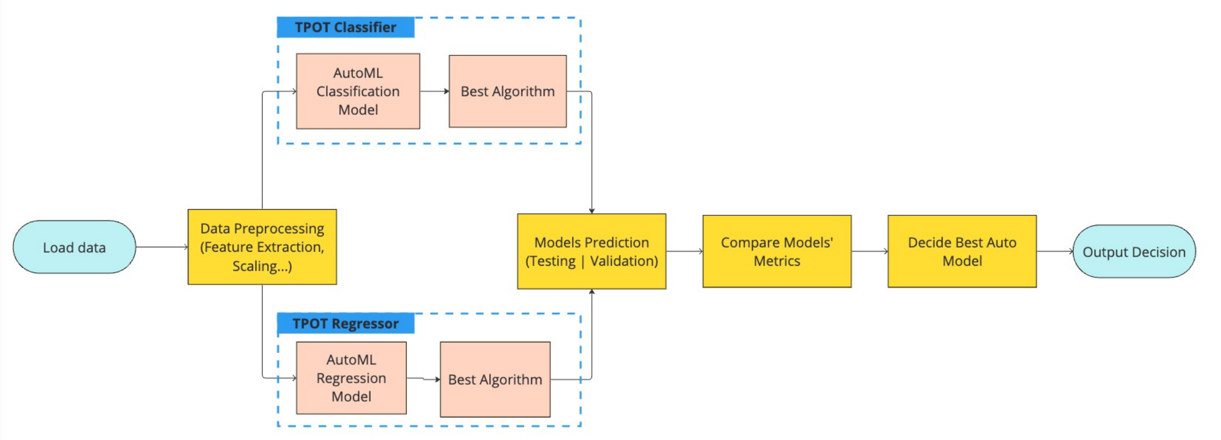



**Figure 6-2: General proposed methodology flowchart.**

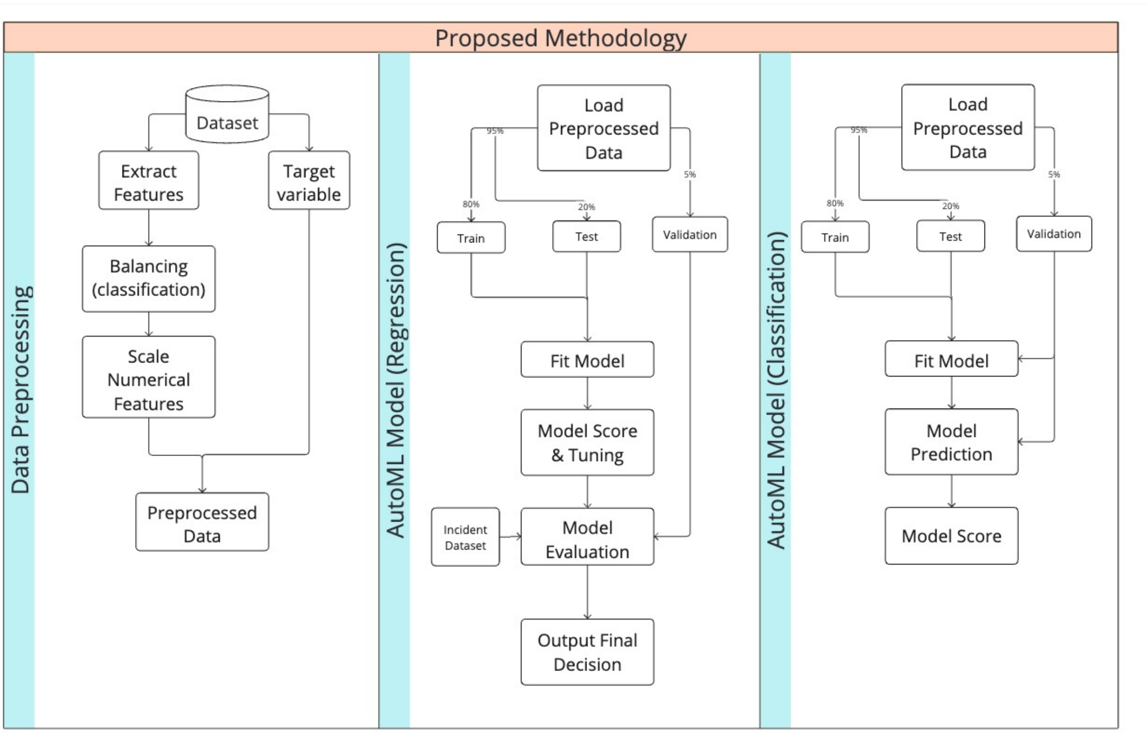



**Figure 6-3: Detailed overview of the modeling phase.**

### *6.3.1 Data Preprocessing*

The proposed methodology involves conducting data preprocessing to prepare the loop detector dataset, which contains traffic variables such as speed, occupancy and flow as a time series, for modeling. This process aims to ensure the accuracy and reliability of ML models in predicting unplanned incidents. The preprocessing primarily involves feature extraction, normalization, and balancing, whenever needed. Normalization is applied to numerical features to scale them to a common range, which is essential for ML models that rely on distance measures. Data balancing is performed to avoid bias towards the majority class, which could result in poor performance when detecting the minority class. Numerous studies have demonstrated that normalization and data balancing significantly improve the performance in various applications, e.g. (Qian & Liu, 2022), (Gain & Hotti, 2021). Algorithms 1 and 2 outline the process of preparing the dataset for modeling by performing data preprocessing for classification and regression task accordingly. This results in a preprocessed dataset that is ready to be used as input for either the classification or regression models.

**Algorithm 1** Data Preprocessing for classification task.

*Input:* dataset (d), output target incidents (t)\

*Output*: Preprocessed dataset (pd)

1. **CleanData:** Data cleaning (d)
2. cd ← CleanData(d)
3. **ExtractFeatures:** Extract feature columns *(more details are described in section 6.1)*
4. f ← ExtractFeatures(cd)
5. **Normalize:** Feature normalization (d)
6. X ← f
7. Y ← t
8. Xn ← normalize(X)
9. **Balance**: Dataset balancing (Xbal,Ybal)
10. Xbal,Ybal ← balance(Xn, Y)
11. **Split:** Splitting dataset into training, validation, and test sets (sd)
12. X_train_val, X_test, y_train_val, y_test ← (Xbal, Ybal, test_size = 0.05)
13. X_train, X_val, y_train, y_val ← (X_train_val, y_train_val, test_size = 0.2)
14. Return pd ← (X_train, y_train, X_val, y_val, X_test, y_test)

**Algorithm 2** Data Preprocessing for regression task.

Input: dataset (d) Output: Preprocessed dataset (pd)

1. **CleanData:** Data cleaning (d)
2. cd ← CleanData(d)
3. **ExtractFeatures:** Extract feature columns *(more details are described in section 6.1)*
4. *f* ← ExtractFeatures(cd)
5. Determine output target: Extract the traffic variable(s) to be predicted (e.g., flow) directly from the input dataset.
6. *t* ← ExtractTarget(d)
7. **Normalize**: Feature normalization (d)
8. *X* ←*f*
9. *Y* ←*t* (where *Y* is the continuous value derived from the dataset)
10. *Xn* ← normalize(X)
11. **Split**: Splitting dataset into training, validation, and test sets (sd)
12. X_train_val, X_test, y_train_val, y_test ← (Xn, Y, test_size = 0.05)
13. X_train, X_val, y_train, y_val ← (X_train_val, y_train_val, test_size = 0.2)
14. Return pd ← (X_train, y_train, X_val, y_val, X_test, y_test)

### 6.3.2 TPOT Models

Our methodology, using TPOT, employ a range of ML techniques and optimization algorithms to understand and adapt to the system's behavior, ultimately enhancing the accuracy and efficiency of the incident detection process. By incorporating the capacity to learn and adapt from past experiences, the models seek to minimize the time and resources required for prediction tasks, leading to cost savings and improved productivity. Central to our approach is the Tree-based Pipeline Optimization Tool (TPOT), an intuitive machine learning library that simplifies the development process, using genetic programming. TPOT's automation extends through various stages of the machine learning workflow, including data pre-processing, model selection, hyperparameter tuning, and ultimately, deployment, all with minimal coding requirements. TPOT is an open-source project on GitHub[8] and an example pipeline is illustrated in Figure 6-4.

[8] https://github.com/rhiever/tpot

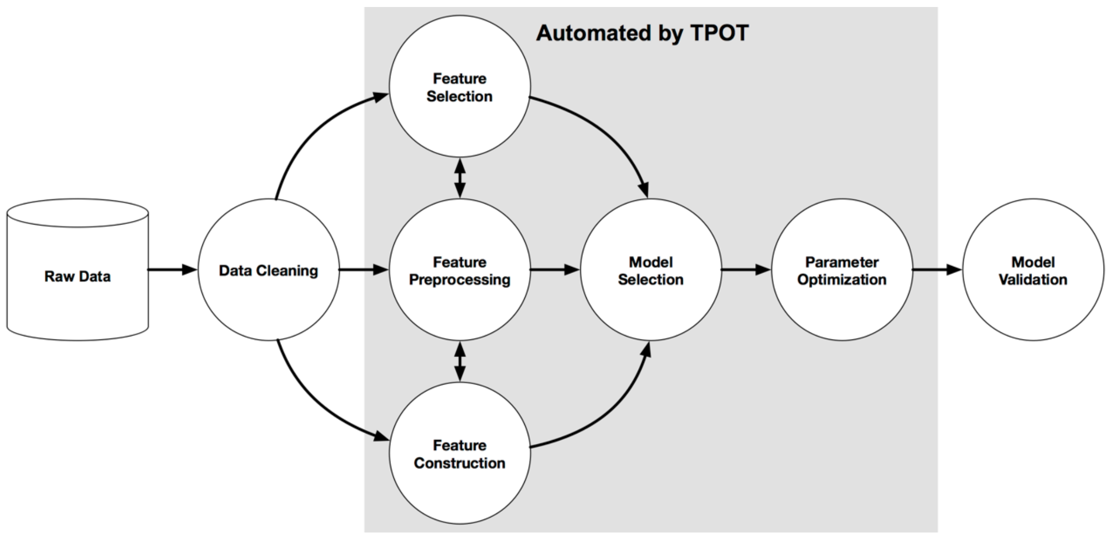



**Figure 6-4: Example pipeline automated by TPOT. (Le, Fu, & Moore, 2020)**

In this research, the power of the TPOT library for constructing and evaluating ML models to predict incidents is demonstrated. For both the classification and regression models, a wide range of advanced ML algorithms have been evaluated automatically. To assess the model's generalization ability, the dataset was divided into three subsets. A portion of 5% was reserved for validation to simulate the model's performance on unseen data. The remaining 95% was then divided into the 80% training set and 15% testing set to ensure the model was trained on a diverse and sufficient dataset. An automated process was employed for selecting the best algorithm and it is provided in the form of pseudocode below as Algorithm 3, which outlines the essential steps for selecting the best model overall, which is based on a comparison of precision (prec), recall (rec), and F1 score (f1).

**Algorithm 3: Best Overall Model Selection in our Methodology**

**Input:** Preprocessed dataset (pd), TPOT Classifier and TPOT Regressor models

**Output:** Best Auto Predictive Detection Model (bAutoD)

1. **Train and evaluate TPOT Classifier:**
2. TPOTClassifier = train_TPOTClassifier(X_train, y_ train)
3. TPOTClassifier_metrics = evaluate_model(TPOTClassifier, X_val, y_val)
4. **Train and evaluate TPOT Regressor:**
5. TPOTRegressor = train_TPOTRegressor(X_train, y_time_train)

```
Predictions = TPOTRegressor.predict(X_val)
Apply threshold to Predictions to categorize as 0 or 1:
Threshold = learn_threshold(y_time_train)
Predictions_binary = apply_threshold(Predictions, Threshold)
TPOTRegressor_metrics = evaluate_model(Predictions_binary, y_class_val)
Model selection based on evaluation metrics (em):
begin
em = (prec, rec, f1)
Best_evaluation_metrics (best_em) = [0, 0, 0, 0, None]
bAutoD= None
for i in range(len(em)):
    if TPOTClassifier_metrics[i] > TPOTRegressor_metrics[i] and
TPOTClassifier_metrics[i] > best_em[i]:
        bAutoD = TPOTClassifier
        best_em[i] = TPOTClassifier_metrics[i]
    elif TPOTRegressor_metrics[i] > best_em[i]:
        bAutoD = TPOTRegressor
        best_em[i] = TPOTRegressor_metrics[i]
end for
Return bAutoD
end
```

## *6.4 The Implementation - Technical Details*

This Section describes the required technical specifications for the development of the proposed subsystem. The components used to build the technical solution are analytically described to offer the reader an overview of the various technical parts.

***Data Ingestion and Preprocessing***

The initial stage involves ingesting traffic data from various sources, including loop detectors, and historical incident reports. The data is ingested in real-time and stored in a scalable data storage solution, specifically in our case Orion Context Broker, to handle the high volume and velocity of incoming traffic data.

Data preprocessing is performed using Python and libraries such as Pandas and NumPy. This step includes cleaning the data to remove noise and inconsistencies, normalizing numerical features to ensure they are on a comparable scale, and balancing the dataset to address class imbalances. Feature extraction techniques are applied to derive meaningful features from the raw data, such as average speed, traffic measurements from upstream and downstream detectors and past timestamps.

### *AutoML Framework*

For model development and selection, the TPOT (Tree-based Pipeline Optimization Tool) tool is utilized. TPOT automates the process of model selection, hyperparameter tuning, and feature engineering using genetic programming. It iteratively explores various machine learning pipelines to identify the most effective model for the given dataset.

The implementation utilizes TPOT's integration with Scikit-learn, allowing the use of a wide range of algorithms and preprocessing techniques. The TPOTClassifier and TPOTRegressor are employed to address the incident detection task from both classification and regression perspective. The training data is split into training and validation sets using Scikit-learn's train_test_split function to evaluate model performance effectively.

### *Model Training and Evaluation*

The training process involves running TPOT to generate multiple candidate models and evaluating them based on predefined performance metrics which have been thoroughly described above. TPOT uses cross-validation to ensure the robustness of the models and prevent overfitting. Once the optimal model is identified, it is further fine-tuned using grid search or random search techniques to optimize hyperparameters. The final model is validated using a separate test dataset to assess its generalization performance.

# 7 Human-in-the-Loop and Explainability in Incident Detection

In this chapter, the integration of human-in-the-loop mechanisms in AI-driven traffic incident detection systems is explored. Human-in-the-loop approaches are essential not only for ensuring that AI models perform accurately but also for ensuring transparency, trust, and confidence among human operators. Explainability plays a crucial role in this process by helping operators understand why AI and data-driven models generate specific predictions about traffic incidents. Through clear explanations of the underlying factors and reasoning behind these predictions, operators are better equipped to provide informed feedback. This feedback allows them to either accept, reject or edit the details of the system-flagged incidents, which thus refines the prediction process. As a result, this dynamic interaction between human oversight and AI systems enhances both the precision and adaptability of traffic incident detection, ensuring the system evolves based on real-world human expertise.

## *7.1 Introduction and Motivation*

The deployment of AI in critical applications such as traffic incident detection necessitates a careful balance between automation and human oversight. While AI models offer unprecedented capabilities in processing and analyzing large datasets to detect incidents, the complexity and opacity of these models often pose challenges in terms of trust and reliability. Human-in-the-loop (HITL) methodologies provide a practical solution to these challenges by incorporating human feedback and ensuring the accuracy of AI predictions. Moreover, the incorporation of explainability features in the system helps enhance the trust in the detection process of AI systems.

The primary motivation for integrating human-in-the-loop mechanisms in AI-driven traffic incident detection systems stems from the need to enhance the accuracy, reliability, and trustworthiness of these systems. By involving human operators in the decision-making process, we can utilize effectively their expertise and contextual understanding to complement the strengths of AI models.

Alongside that, another significant aspect of human-in-the-loop is the ability to provide immediate feedback on AI predictions. When an incident is identified by the system, the operator is prompted to acknowledge the incident, confirming its occurrence. This feedback loop ensures that false positives are minimized and that the system's predictions align with real-world scenarios. Additionally, if an incident occurs and the system fails to report it, operators can manually insert this information, ensuring that critical events are not overlooked. This two-way interaction not only improves the system's accuracy but also provides valuable data for retraining and refining the AI models over time.

Another critical motivation is the need for explainability and trustworthiness in AI systems. Traffic incident detection is a high-stakes application where the decisions made by AI models can have significant implications for public safety and urban management. Integrating techniques such as LIME (Local Interpretable Model-agnostic Explanations) and SHAP (SHapley Additive exPlanations) allows us to provide transparent and understandable predictions. These techniques help uncover the reasoning behind AI decisions, making it easier for human operators to trust and rely on the system. By ensuring that AI predictions are not only accurate but also explainable, we can instill greater confidence in the adoption of these technologies in real-world settings.

This chapter aims to provide an exploration of human-in-the-loop methodologies in the context of AI-driven traffic incident detection. It begins with a detailed discussion on the mechanisms for incorporating human feedback into AI predictions, in various domains and more specifically, in the transportation sector. Furthermore, the chapter delves into the importance of explainability and trustworthiness in AI systems, focusing specifically on the incident detection task. An overview of techniques such as LIME and SHAP are provided, along with a detailed explanation of how these methods are integrated into the deployed AI models, and the impact of explainable AI on operator trust and decision-making processes is thoroughly presented. Finally, the chapter explores potential advancements in human-in-the-loop methodologies and discuss the integration of advanced AI techniques with human feedback for enhanced traffic incident detection.

Regarding the evaluation of the impact of human feedback and explainability on system performance, which includes assessing system performance metrics before and after the integration of human-in-the-loop mechanisms and conducting a comparative study of AI models performance metrics, this is presented as part of the real-world case studies in Chapter 9.

By addressing these areas, this chapter aims to provide a thorough understanding of the critical role that human-in-the-loop methodologies play in enhancing the effectiveness and trustworthiness of AI-driven traffic incident detection systems.

## *7.2 Human-in-the-Loop State-of-the-art*

### *7.2.1 Human-in-the-loop in ML*

Human-in-the-loop (HITL) approaches in machine learning combine human intelligence with automated systems to overcome challenges related to model performance, data limitations, and interpretability. The integration of human feedback allows for models to improve incrementally through interventions during the training process, ensuring higher accuracy and trustworthiness in decision-making. HITL is increasingly significant in fields such as natural language processing, computer vision, and intelligent transportation systems (Wu, et al., 2022).

A typical ML framework with Human-in-the-loop (HITL) learning is shown in Figure 7-1, which consists of three components: data pre-processing, data modeling, and modifying the process to improve performance (Kumar, et al., 2024)

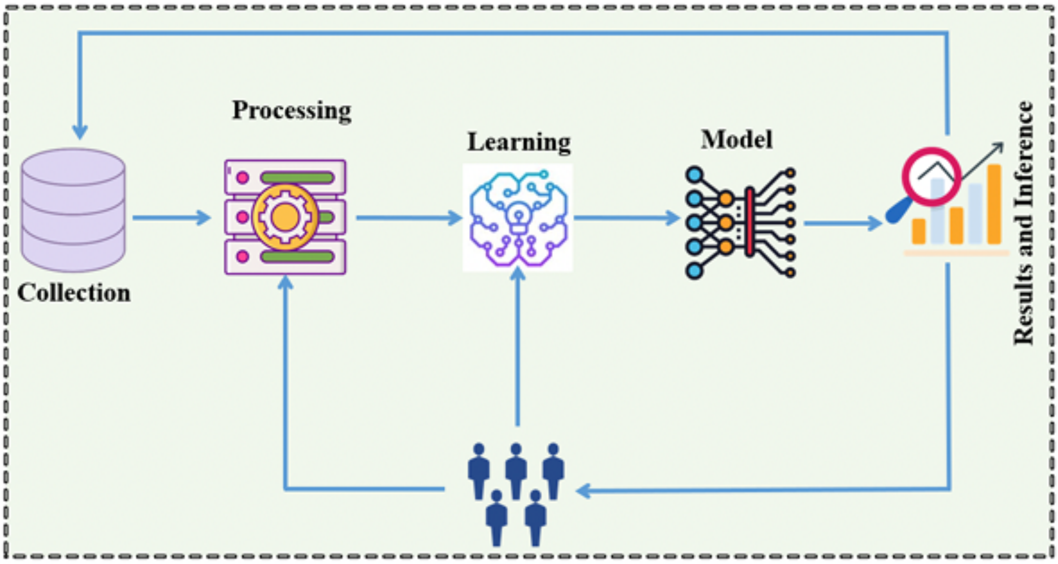



**Figure 7-1: Human-in-the-loop learning framework. (Kumar, et al., 2024)**

The HITL offers several advantages (Kumar, et al., 2024), such as:

- **Improved performance:** As people validate or reject, thus interact with, the model's answers to various events, the algorithm improves accuracy and consistency.
- **Improved data acquisition:** HITLs can create and assure accurate data for ML models in data-scarce scenarios.
- **Bias handling:** Human-designed AI algorithms can perpetuate inequality, while HITL can detect and fix bias early on.
- **Increased efficiency:** While not all components of the process may be automated, a large number of them can, resulting in saving time and financial resources.

### *7.2.2 Human-in-the-loop in ML Literature Review*

Researchers are defining new types of interactions between humans and machine learning algorithms, which we can group under the umbrella term of Human-in-the-loop machine learning (HITL-ML) (Munro, 2020).The idea is not only to make machine learning more accurate or to obtain the desired accuracy faster, but also to make humans more effective and more efficient. Depending on who is in control of the learning process, we can identify different approaches to HITL-ML (Holmberg, Davidsson, & Linde, 2020)

- **Active learning** (AL) (Settles, 2009), in which the system remains in control of the learning process and treats humans as oracles to annotate unlabeled data.
- **Interactive machine learning** (IML) (Amershi, Cakmak, & Knox, 2014), in which there is a closer interaction between users and learning systems, with people interactively supplying information in a more focused, frequent, and incremental way compared to traditional machine learning.
- **Machine teaching** (MT) (Simard, Amershi, & Chickering, 2017), where human domain experts have control over the learning process by delimiting the knowledge that they intend to transfer to the machine learning model.

One of the most prominent uses of HITL systems is in data processing, where the need for large, annotated datasets presents challenges due to the high cost of labeling. HITL frameworks optimize this process through iterative labeling and active learning, where human annotators focus on the most challenging samples. This has proven effective in improving the overall quality of the data fed into machine learning models (Yu et al., 2015) .Researchers such as (Liu, Feng, & Wang, 2021)have demonstrated the effectiveness of HITL in improving object detection tasks through human-assisted annotation processes. These iterative approaches minimize errors by having humans intervene when the model cannot confidently label data. Similar approaches have been applied to natural language processing (NLP), where HITL aids in tasks like sentiment analysis and question-answering by providing critical feedback on model predictions (Liu, Feng, & Wang, 2021)

In model training, HITL frameworks are employed to incorporate human judgment, especially in tasks where the model is prone to make errors. This dynamic interaction allows human operators to correct predictions or guide the model toward better generalization. HITL frameworks have been highly effective in text classification and semantic parsing tasks, where the ambiguity of language data often requires human oversight to resolve. Explainability is one of the key features of HITL systems. By involving human experts, these systems can offer insights into why models make certain decisions. For instance, (Arous, et al., 2021) proposed a human-AI hybrid model that improves both the performance and explainability of text classification models. This approach helps build trust in AI systems by allowing humans to correct models when necessary and understand their decision-making processes (Arous, et al., 2021). In computer vision, HITL systems enhance performance in tasks such as image segmentation, video object tracking, and object detection. The inclusion of human feedback ensures that models refine their predictions in cases of occluded or blurred objects. Studies by Madono et al. (2020) have shown significant improvements in the recall rate of object detection tasks by integrating human-in-the-loop strategies (Madono, Nakano, Kobayashi, & Ogawa, 2020).

The adoption of HITL frameworks is happening in various domains, each getting substantial benefits from the combination of human judgment and machine learning algorithms. In security systems, for instance, HITL plays a critical role in ensuring accurate decision-making in safety-critical environments, such as nuclear power plants and commercial aviation. Singh and Mahmoud (2020) highlighted how HITL systems help avoid catastrophic errors by allowing human operators to intervene when necessary. Their work focused on using human feedback to improve system safety in complex industrial settings (Sing & Mahmoud, 2020). Similarly, in software engineering, HITL systems have been applied to code debugging and program repair. MacHiry et al. (2013) developed Dynodroid, a HITL-based system that improves Android app testing by allowing humans to provide feedback during event-driven program analysis. This system allows for greater accuracy in identifying bugs and potential vulnerabilities (Machiry, Tahiliani, & Naik, 2013). In simulation systems, HITL frameworks are employed in process optimization and decision-making, with applications in logistics, medical diagnostics, and traffic management. Demirel et al. (2020) highlighted the advantages of incorporating human expertise into simulation models for more accurate forecasts and strategic planning. (Demirel, 2020)

While HITL systems offer numerous benefits, several challenges remain. One of the primary concerns is scalability. As machine learning models grow more complex, the need for human intervention may become overwhelming. Researchers are exploring ways to optimize feedback loops, using techniques like active learning to prioritize the most critical human interactions (Liu, Feng, & Wang, 2021).Moreover, there is a need to develop more user-friendly interfaces that allow non-expert users to interact with HITL systems effectively. Current frameworks often rely on domain experts to provide meaningful feedback, limiting the accessibility of these systems in broader applications. Future research should focus on creating intuitive, easily navigable interfaces to democratize the use of HITL systems (Zhang, He, Dragut, & Vucetic, 2019).

All in all, human-in-the-loop systems offer a promising approach to overcoming the limitations of fully automated machine learning models. By integrating human expertise into various stages of data processing, model training, and system applications, HITL frameworks ensure more accurate, explainable, and trustworthy AI systems.

### *7.2.3 Human-in-the-loop in Transportation*

In transportation systems, machine learning methods with the concept of inclusion of humans, such as online, stochastic, and offline learning, are critical for real-time data processing and decision-making in order to adapt to fast changing data.

- Online learning continuously updates models with new data, allowing traffic management systems to respond instantly to changing conditions. HITL ensures that human feedback is incorporated, correcting model errors and refining outputs for better decision-making in dynamic environments.
- Stochastic learning, including methods like Stochastic Gradient Descent (SGD), allows incremental model updates using small data batches. This method optimizes computational efficiency, while human interaction helps guide

updates, making transportation systems adaptable to evolving traffic conditions.

- Offline retraining periodically updates models with new data, triggered either by set time intervals or performance drops. Human involvement is vital in this case, as it fine-tunes the retraining process, ensuring that models adapt effectively without too many resources.

The integration of human feedback through HITL across these learning methods ensures that transportation systems remain flexible, efficient, and responsive, optimizing traffic incident detection and management in real-time environments.

Online learning enables ML models to update incrementally with new data, making it highly effective in dynamic environments like traffic incident detection. By continuously adapting to new information, online learning ensures real-time responsiveness and scalability without retraining from scratch. In the HITL context, humans provide critical feedback to fine-tune these models, ensuring that the updates align with real-world complexities and improving decision-making accuracy over time. The iterative interaction between humans and the model is essential in online learning tasks, such as streaming data analytics or time-series predictions. By combining HITL with stochastic learning methods like Stochastic Gradient Descent (SGD), online learning processes data sequentially, enabling fast and scalable model updates with human oversight. This hybrid approach ensures optimal performance in dynamic environments while maintaining efficiency. For long-term accuracy, offline model retraining can be periodically triggered or set based on performance thresholds, keeping the models up to date without overburdening computational resources.

There exist few works which include HITL techniques and approaches in the transportation sector. Specifically, one study by Chiang et al. 2010 presents a hierarchical longitudinal automation system designed to ensure safe and comfortable vehicle operations through HITL integration. This system employs an adaptive detection area that processes sensor data for vehicle detection, particularly on curves.

Supervisory control utilizes this data to calculate desired velocities for smooth and safe operation across different modes, while regulation control leverages soft-computing techniques to execute velocity commands effectively. (Chiang, Wu, Perng, Wu, & Lee, 2010) Similarly, HITL principles have been explored in civil infrastructure inspection, where automation-assisted technologies, such as drones and underwater vehicles, leverage human expertise to improve efficiency and safety. A review by Agnisarman et al. 2019 regarding automated visual inspection methods highlights how HITL systems reduce inspector bias, augment qualitative assessments, and minimize exposure to hazardous environments. However, studies emphasize the need for further research on human factors, including cognitive demands, trust, and communication, to optimize these systems for seamless collaboration. (Agnisarman, Lopes, Madathil, Piratla, & Gramopadhye, 2019)

Further innovations in HITL methodologies have emerged in frameworks designed for autonomous vehicles (AVs) operating in mixed traffic environments. The Human as AI Mentor-based Deep Reinforcement Learning (HAIM-DRL) framework exemplifies this approach, integrating human expertise into reinforcement learning to improve safety and traffic flow efficiency. In this framework, human mentors guide AI agents by intervening in high-risk situations and demonstrating appropriate actions to prevent accidents. Comparative analyses reveal superior performance in safety, traffic flow optimization, and adaptability to novel scenarios, underscoring the transformative potential of HITL approaches. (Huang, Sheng, Ma, & Sikai Chen, 2024) Together, these studies highlight how HITL frameworks are redefining automation in transportation and infrastructure, ensuring human-centric solutions that balance technological advancement with practical implementation.

## 7.3 Explainability State-of-the-art

Artificial Intelligence has seen exponential growth over the last decade, however, along with the rapid advancements comes a growing need for transparency, accountability, and trust in AI systems. This necessity has given rise to the field of Explainable AI (XAI), which seeks to make AI systems more interpretable and their decision-making processes understandable to humans. Unlike traditional AI models that function as "black boxes," XAI aims to provide insights into how an AI system reaches a particular conclusion, facilitating better human-AI interaction and fostering trust among users and stakeholders.

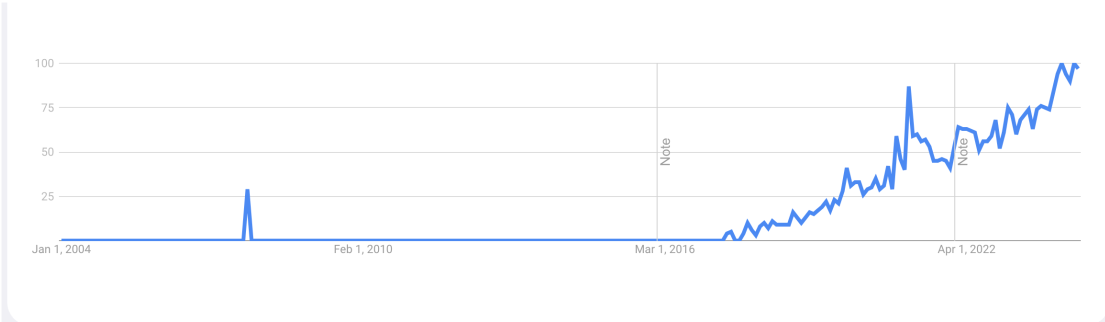



**Figure 7-2: Google trends of the term explainable ai over the last 10 years.**

The significance of XAI can be reflected in its growing popularity and increasing demand within both academia and industry. As shown in the Google Trends analysis of the term "Explainable AI" over the last 10 years (Figure 7-2) , there has been a notable rise in interest starting around 2017, with steep growth observed since then. This is likely due to the increasing deployment of AI systems in sensitive domains such as autonomous driving, where understanding the reasoning behind AI-generated predictions is crucial for decision-makers. The graph highlights how awareness and discussion of explainability in AI have significantly intensified, reflecting a broader societal and technical shift towards responsible AI development. As AI technologies became more complex and widely adopted, concerns around fairness, ethics, and bias also have emerged. Public trust in AI systems became directly tied to how well those systems could explain their actions. As a result, explainable AI has gained attention, not only as a research priority but also as a regulatory concern, with many institutions now requiring AI systems to be transparent and interpretable.

### *7.3.1 Explainability Advantages*

Explainable AI (XAI) represents a breakthrough in the way we interact with and understand decisions made by artificial intelligence systems. Whereas traditional AI often functions as a "black box," providing results without explaining the process that generated them, Explainable AI aims to make these processes transparent and understandable to humans. This approach not only increases trust and acceptance of AI systems but also provides human operators with the tools needed for effective supervision and intervention. Some of the various aspects in which Explainable AI adds value to human-machine integration are listed below (Minh, Wang, & Li, 2022):

1. **Transparency and Understanding**: Explainable AI provides insights into the "how" and "why" behind decisions made by AI. This helps human operators understand the underlying patterns behind the decision-making processes, making it easier to identify and correct any errors or biases in the system.

2. **Trust and Accountability**: When users and supervisors understand AI processes, they are more likely to trust its decisions. This is especially important in critical areas such as medicine, security, and law, where trust is a key factor.

3. **Improved Human – AI Interaction**: Explainability facilitates more effective collaboration between humans and machines. Operators can use the information provided by Explainable AI to make informed decisions, taking full advantage of AI's data analysis capabilities and human intuition.

4. **Legal and Ethical Compliance**: In many industries, transparency and accountability are not only ethical expectations but also legal requirements. Explainable AI can help meet these requirements by providing clear and documentable explanations of decisions.

5. **Feedback and Continuous Learning**: The ability to understand AI decisions enables operators to provide more accurate feedback, which can be used to improve and refine AI models. This feedback loop contributes to continuous improvement of systems.

### *7.3.2 Explainability in ML*

Explainability provides insight into the AI models' decision to the end-user in order to build trust that the system is making correct and non-biased decisions based on facts. Figure 7-3 depicts the distinction between white-box, gray-box, and black-box decision-making processes, as well as shows how explainable AI (XAI) is applied to achieve a trustworthy model with a good interpretability-accuracy tradeoff (Ali, et al., 2023).

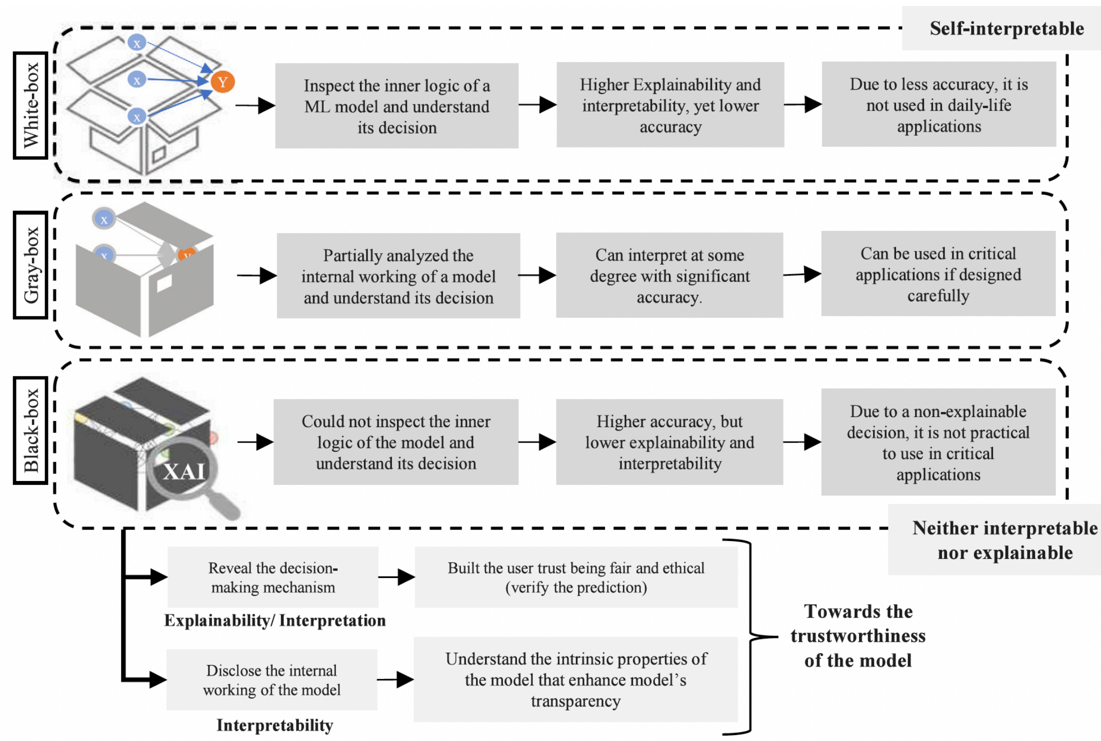



**Figure 7-3: Distinction between white-box, gray-box, and black-box decision-making processes (Ali, et al., 2023).**

The primary objective of research in Explainable AI (XAI) is to enhance the comprehensibility and transparency of AI systems for humans without compromising their performance. The ability to detect hidden patterns in complex data is both an advantage and a limitation: while AI models can automatically uncover intricate structures in data, these learned patterns often remain obscured, with no explicit rules or logical processes involved, especially in Deep Learning algorithms. Although AI algorithms can identify correlations across diverse and complicated datasets, there is no guarantee that these correlations are meaningful or reflect actual causal relationships. (Rieg, Frick, Baumgartl, & Buettner, 2020). Additionally, the complexity of models, especially advanced deep neural networks (DNNs), frequently prevents human operators from easily inspecting or controlling them. As such, AI presents both opportunities for innovation and challenges related to security, safety, privacy, and transparency.

XAI aims to produce human-interpretable models, especially for high-stakes sectors like the military, banking, and healthcare, where domain experts require not only effective problem-solving tools but also meaningful explanations to trust and understand the results. These interpretable outputs are valuable not only for experts to validate decisions but also for developers to investigate potential errors in the system. AI methods facilitate (i) assessing current knowledge, (ii) advancing it, and (iii) developing new assumptions or theories. The goals of XAI include enhancing justification, control, improvement, and discovery in AI models. Key benefits of making these "black-box" systems more transparent include (Guidotti, et al., 2018):

- Empowering users to mitigate negative consequences of automated decision-making.
- Assisting individuals in making more informed choices.
- Uncovering and addressing security vulnerabilities.
- Aligning algorithms with human values.
- Improving industry standards for AI development, boosting consumer and business confidence.
- Supporting the enforcement of the Right of Explanation policy.

For an AI model to gain acceptance from end-users and industries, it must be trustworthy (Véliz, Prunkl, Phillips-Brown, & Lechterman, 2021). Achieving this trust, however, is challenging. Factors contributing to trustworthiness include fairness (Mehrabi, Morstatter, Saxena, Lerman, & Galstyan, 2021), robustness (Oberman, 2021), interpretability (Li, et al., 2022), and explainability (Das & Rad, 2020). Explainability, in particular, is a crucial element. Current research largely focuses on improving explanations in addition to providing insights for future work, with researchers proposing different methods for explaining AI models using natural language, mathematical descriptions (Ribeiro, Singh, & Guestrin, 2016), or visualizations (Doshi-Velez & Kim, 2017).

### *7.3.3 Explainability in Intelligent Transportation Systems*

Transportation systems, such as intelligent transport systems (ITS) and autonomous vehicles, rely on advanced machine learning models to make critical decisions. Explainability is crucial in such systems, particularly due to the high-stakes nature of decisions in the field of transportation, where public safety is involved. Sahoo and Mohan's work highlights the need for explainability in ITS, noting that as these systems grow in complexity, it becomes essential to provide transparency to operators and users to build trust and accountability (Sahoo & Mohan, 2022) The authors emphasize that explainability allows for error diagnostics, system resilience, and the mitigation of biases, which are key for the smooth functioning of transportation infrastructures. (Adadi & Bouhoute, 2023).

Various explainability methods have been developed to address the opaque nature of AI models in transportation. According to the literature, explainability can be achieved through several approaches:

- **Model-agnostic techniques** such as Local Interpretable Model-agnostic Explanations (LIME) are widely used to provide post-hoc explanations for model predictions. (Adadi & Bouhoute, 2023) (Olugbade, Ojo, Imoize, Isabona, & Alaba, 2022). These methods do not require knowledge of the internal workings of the model and instead provide local explanations for individual predictions.

- **Surrogate models** that mimic the behavior of more complex systems are another popular technique. These interpretable models offer a simplified representation of the AI system, making it easier for human operators to understand the decision-making process. (Adadi & Bouhoute, 2023)

Explainable AI has been applied in various transportation sectors, including incident detection, traffic management, and autonomous driving. For instance, (Olugbade, Ojo, Imoize, Isabona, & Alaba, 2022)discuss how AI-based incident detection systems rely on explainability to monitor traffic and manage road incidents more effectively. These systems often use sensors, video feeds, and other data sources to predict and detect traffic anomalies. However, without adequate explanations, operators may struggle to trust or act upon the AI-generated insights, making post-hoc explainability a critical component for deployment in real-world scenarios.

Similarly, Sahoo and Mohan explore how explainable AI can be applied to improve the safety and predictability of autonomous vehicle systems. (Sahoo & Mohan, 2022) By providing human interpretable feedback, XAI enables more precise control over vehicle behaviors, such as lane departure warnings and adaptive cruise control systems.

Despite the advancements in XAI techniques, challenges remain. One major concern is balancing transparency and performance. Complex AI models, particularly deep learning algorithms, often provide higher predictive accuracy but are more difficult to explain. Simplifying such models could lead to a loss in performance, thus affecting the reliability of the system. (Sahoo & Mohan, 2022) Future research in XAI for transportation will likely focus on improving the interpretability of increasingly complex AI models without compromising their performance. There is also a growing emphasis on developing domain-specific explanation techniques tailored to the needs of transportation operators, regulators, and end-users.

## *7.4 Proposed methodology*

The proposed framework for enhancing incident detection systems integrates both Human-in-the-Loop (HITL) methodologies and explainability features, combining the strengths of artificial intelligence (AI) with human expertise to ensure robust, accurate, and reliable performance. By leveraging HITL, the system allows for continuous human intervention and oversight, enabling experts to validate and refine AI-generated outputs. This hybrid approach ensures that the system can dynamically learn from human feedback while providing interpretable insights through explainability features. These explainability mechanisms are critical for fostering trust in AI-driven systems, as they allow human operators to understand the rationale behind AI decisions, diagnose potential errors, and make adjustments to improve system accuracy. Ultimately, this framework aims to enhance the detection of non-recurring incidents, such as traffic disruptions or accidents, while maintaining high levels of transparency, trust, and operational efficiency in real-world transport systems.

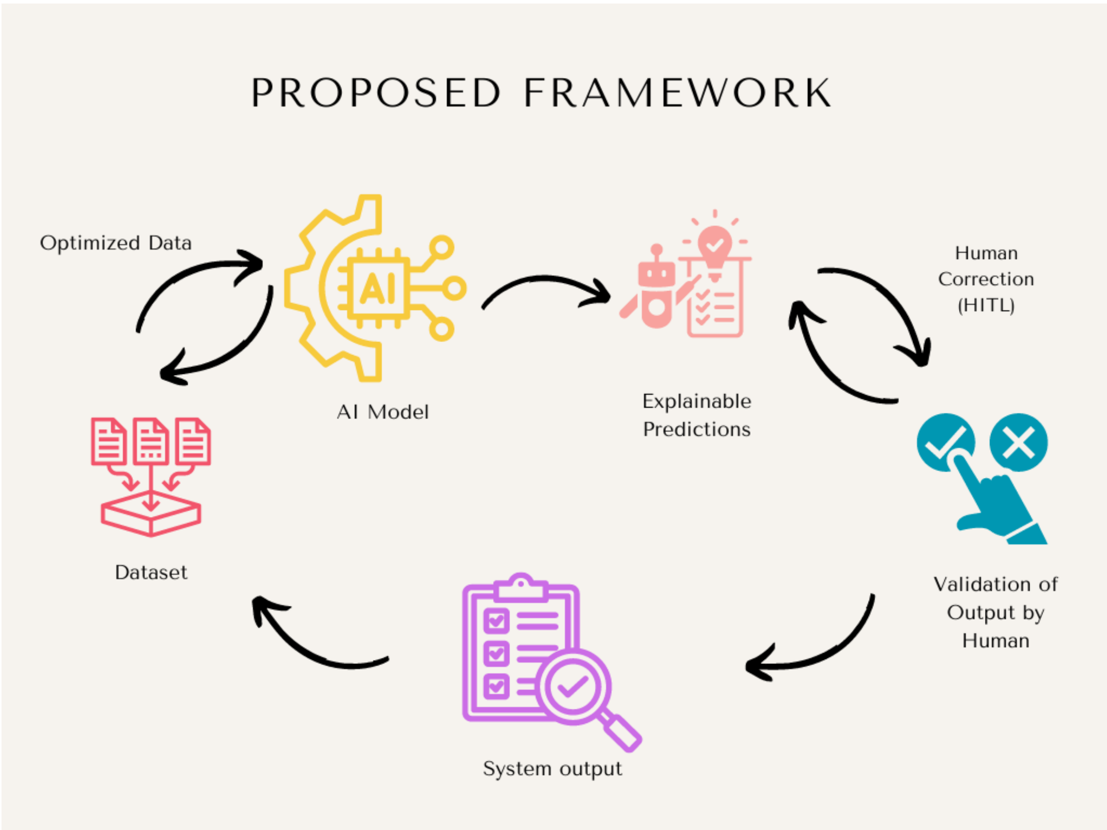



**Figure 7-4: Our proposed framework for integrating Explainability and Human-in-the-Loop approaches.**

The key components of the proposed framework are depicted in Figure 7-4 and described as follows:

***1. Dataset***

The foundation of the incident detection system is a comprehensive dataset comprising traffic data collected from various sources, such as loop detectors, segment-level measurements, and historical incident records. This dataset is pre-processed to remove noise and normalized to ensure consistency, making it suitable for training AI models.

***2. AI Model***

The core of the framework is the most appropriate AI model, which is built and selected using advanced machine learning algorithms. This model is trained initially on the dataset and later on the optimized dataset to learn patterns and make predictions about potential incidents. The choice of algorithm is based on the specific requirements and characteristics of the traffic data, ensuring the model is well-suited for real-time incident detection.

***3. Explainable Predictions***

To ensure transparency and trust in the AI model's decisions, the framework incorporates explainability features. Techniques such as SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-agnostic Explanations) are used to provide insights into the model's predictions. These tools help stakeholders understand which features influenced the predicted values, making the system's outputs more interpretable and reliable.

***4. Human-in-the-Loop (HITL)***

Human-in-the-Loop (HITL) methodologies are integrated to leverage human expertise in refining the model's predictions. Traffic management professionals review the explainable predictions generated by the model and provide corrections or adjustments as necessary. This iterative feedback loop ensures that the model continuously learns from human input, improving its accuracy over time.

***5. Validation of Output by Human***

Following human correction, the system's output is validated by human experts to ensure its reliability and accuracy. This validation step is crucial for identifying and rectifying any potential errors or biases in the AI model's predictions. By involving human oversight, the system maintains high standards of performance and trustworthiness.

***6. System Output***

The final validated output is then generated by the system, providing actionable insights and alerts to traffic management personnel. This output includes real-time incident detection notifications and prompts for mitigating traffic issues, ensuring a timely and effective response to incidents.

***7. Continuous Improvement Cycle***

The validated system output, along with the human corrections and validations, is fed back into the system, creating a continuous improvement cycle. This cycle allows the AI model to learn from new data and human feedback, continuously enhancing its performance and adaptability to evolving traffic conditions.

There are several benefits of the proposed framework, some of which are listed below:

- **Enhanced Accuracy and Reliability:** By integrating human expertise through HITL methodologies, the system minimizes errors and improves the reliability of incident detection.
- **Increased Transparency and Trust:** Explainability features provide clear insights into the AI model’s decision-making process, making it easier for stakeholders to understand and trust the system’s outputs.
- **Continuous Learning and Adaptation:** The continuous improvement cycle ensures that the AI model evolves with changing traffic patterns and incorporates the latest data and human insights. This adaptability is essential

for maintaining high performance in dynamic and complex traffic environments.

- **Proactive Traffic Management:** The system's ability to provide real-time, validated incident detection enables traffic management personnel to respond proactively to incidents. This proactive approach helps in mitigating traffic congestion and improving overall road safety.

In summary, the proposed framework for integrating Human-in-the-Loop approaches and explainability features in incident detection systems offers a robust solution for enhancing traffic management. By integrating human expertise into the machine learning workflow, our research seeks to address challenges such as sensor noise, data sparsity, and the inherent unpredictability of traffic incidents. This approach not only enhances the performance of the models but also ensures that the system remains adaptable and resilient in real-world applications. The combination of the strengths of AI with human expertise and transparent decision-making ensures that the system detects accurately and reliably emerging situations, ultimately contributing to safer and more efficient traffic management.

### *7.4.1 Integration of Explainability features*

In the context of our research, the transparency and interpretability of machine learning models are paramount. While many models can achieve high accuracy, their "black-box" nature often makes it difficult to understand how they arrive at specific predictions. This lack of transparency can be a significant barrier in fields where trust and accountability are crucial, including the task of automatic detection of incidents. In this section, we present the integration of explainability tools and approaches used in the context of the aforementioned general methodology.

One of the methods used is SHAP (SHapley Additive exPlanations), being a unified approach to interpreting the output of machine learning models. It is based on cooperative game theory, where each feature in a dataset is treated as a "player" in a game, and the model's prediction is the "payout" that needs to be fairly distributed among the features. The SHAP values quantify the contribution of each feature to the final prediction, making it possible to decompose the prediction into the sum of the contributions from individual features.

We have selected SHAP because it provides a theoretically sound and consistent method to explain individual predictions of complex models. SHAP offers the following advantages:

- **Model-agnostic**: SHAP can be applied to any machine learning model, making it versatile across different types of models.
- **Local and global explanations**: SHAP values can explain individual predictions (local explanations) and provide insights into the overall behavior of the model (global explanations).
- **Fairness and consistency**: SHAP is grounded in Shapley values from cooperative game theory, ensuring that the contributions of features are fairly distributed based on their actual impact on the prediction.

On the other hand, LIME (Local Interpretable Model-agnostic Explanations) is a technique designed to explain the predictions of any machine learning model by approximating it locally with an interpretable model. The core idea behind LIME is to understand the model's predictions by perturbing the input data and observing the resulting changes in predictions. This allows us to build a local, interpretable model (like a linear model or decision tree) that can explain the predictions in the vicinity of the instance being analyzed.

The motivation for choosing LIME in our analysis stems from the need for interpretability in machine learning models, particularly in understanding how specific predictions are made. Specifically, we selected LIME because it offers:

- **Model-agnostic explanations**: LIME can be applied to any machine learning model, regardless of its complexity or architecture.

- **Local interpretability**: LIME focuses on explaining individual predictions, making it possible to understand the model's behaviour in specific instances, which is especially useful for case-by-case analysis.

- **Simplicity and Flexibility**: By fitting an interpretable model locally around the prediction, LIME provides straightforward explanations that can be easily understood and communicated.

Figure 7-5 provides a detailed overview of the comprehensive XAI workflow developed as part of our research. The workflow is divided into three primary stages: Data Preparation, Model Development, and Comprehensive Analysis.

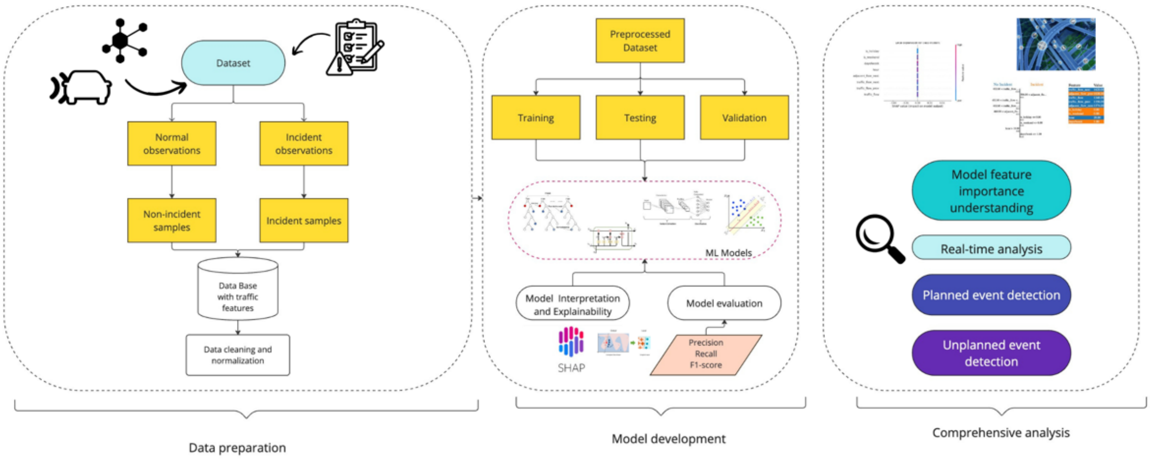



**Figure 7-5: Comprehensive workflow for proposed XAI framework, including 3 primary stages.**

In the *Data Preparation* stage, the dataset is curated by collecting data from loop sensors, which is then categorized into normal observations and incident observations. These categories are further divided into non-incident and incident samples, respectively. Following this categorization, data cleaning and normalization procedures are applied to ensure the dataset is prepared for model development.

The *Model Development* stage follows, where the preprocessed dataset is split into training, testing, and validation sets. Various machine learning methodologies are applied to these datasets, focusing on model interpretability and explainability, particularly using SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable

Model-agnostic Explanations). The models are then evaluated based on their precision, recall, and F1-score to ensure their effectiveness in predicting incidents.

The final stage, *Comprehensive Analysis*, involves a deeper examination of the model's performance. This includes understanding model feature importance, conducting analyses for explaining local and global predictions, and detecting events within the transport system. The insights gained from this stage are crucial for refining the models and ensuring their applicability in real-world scenarios, ultimately contributing to the development of a more reliable and efficient intelligent transport system.

This structured approach allows our research to systematically address the challenges of incident prediction in intelligent transport systems, ensuring that the models developed are both accurate and interpretable, with a strong emphasis on real-world applicability.

### *7.4.2 Integration of Human Feedback*

Figure 7-6 illustrates the Human-in-the-Loop (HITL) framework, a fundamental methodology utilized in our research for enhancing machine learning (ML) models designed to predict incidents within intelligent transport systems using loop sensor data. The process initiates with the collection of data from loop sensors embedded within roadways, which is then employed to train ML models aimed at incident prediction. Rather than depending solely on automated predictions, the HITL approach introduces a critical intermediate step wherein human experts review and validate these predictions.

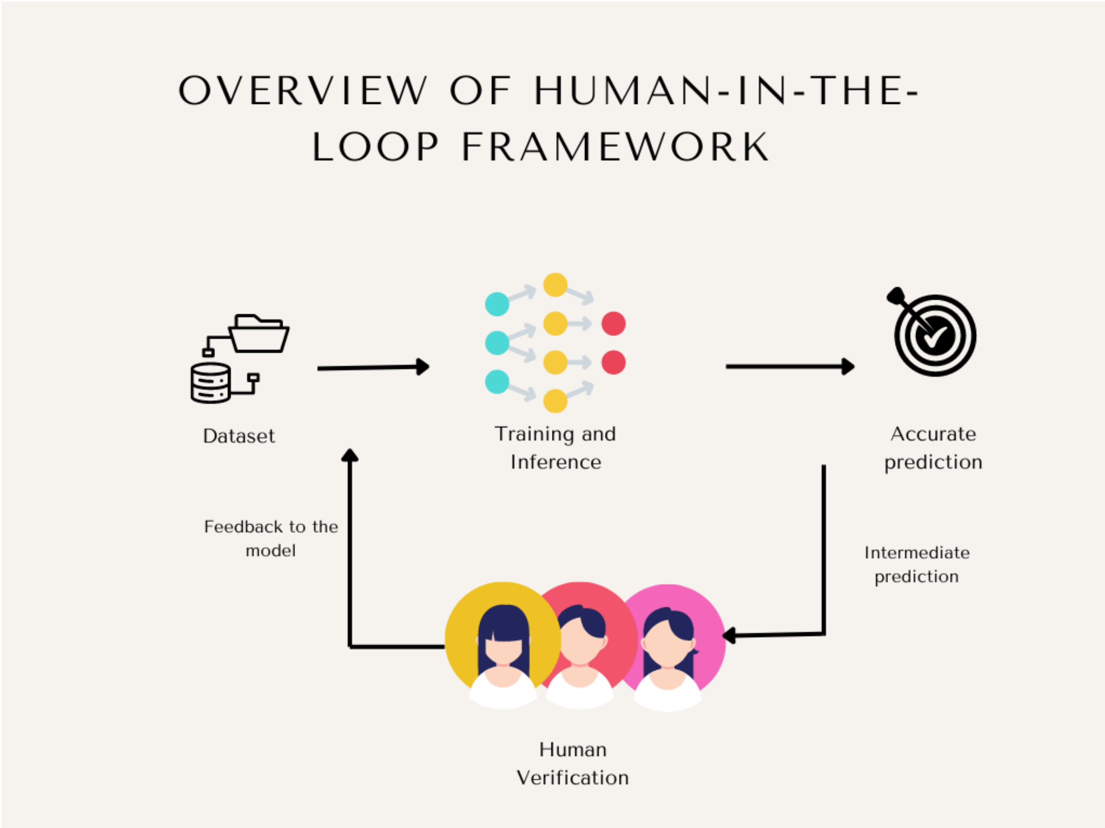



**Figure 7-6: Overview of methodology specifically for Human-in-the-Loop approach.**

The feedback provided by these experts is then reintegrated into the model, further refining its accuracy and allowing the model to adapt to the complexities encountered in real-world scenarios. This iterative process ensures that the ML models not only improve progressively but also align more closely with the nuanced understanding provided by transport professionals, ultimately leading to more precise and reliable predictions of incidents within intelligent transport systems.

Integrating human feedback into ML models provides a powerful mechanism for continuous improvement in AI-driven traffic incident detection systems. This integration can be broken down into several key processes:

1. **Incident Acknowledgment and Correction:** When the AI system detects a traffic incident, it prompts the human operator to acknowledge the detection. This confirmation serves as a validation step, ensuring that false positives are minimized. Conversely, if an incident occurs that the AI system fails to detect,

operators can manually report it, providing critical data for retraining the model to recognize similar incidents in the future.

2. **Incorporating Feedback into Model Updates:** The feedback from human operators is incorporated into the online learning algorithm. For instance, when an operator confirms or corrects an incident detection, this feedback is used to adjust the model parameters, enhancing its accuracy and responsiveness. This continuous feedback loop ensures that the AI system evolves and improves over time.

3. **Improving Model Trust:** By incorporating human feedback, the AI system can learn from real-world scenarios that may not be well-represented in the initial training data. This iterative learning process helps capture a broader range of traffic incident types and conditions, enhancing the model's robustness.

# 8 Information System AutoEventX

In this Chapter, the developed AI-driven information system is explained. The developed information system is called AutoEventX and incorporates the functionalities presented in Chapter 4, Chapter 5, Chapter 6 and Chapter 7. The implemented information system is evaluated and deployed in real-world case studies as described in Chapter 9.

## *8.1 System architecture and implementation*

The conceptual architecture of the proposed system is illustrated in Figure 8-1.

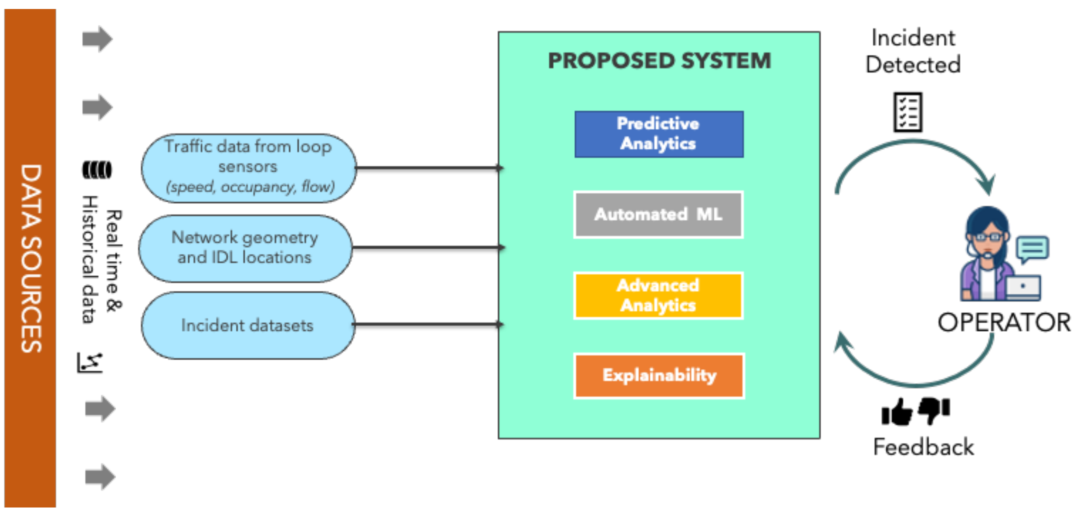



**Figure 8-1: Conceptual Architecture of the Proposed System.**

The overall information system is addressed through the integration of different tools and services addressing the various phases of the proposed framework. The system is able to collect and fuse data provided by different sources, to evaluate the quality of the dataset, through dedicated techniques especially regarding the measurements captured by loop detectors, to support efficient real-time data processing, to provide stakeholders with predictions regarding planned and unplanned events and the rationale behind these predicted results while taking into account the expert operators' feedback, This is in combination with a large amount of historical data and taking into account background domain knowledge.

### *8.1.1 Data Processing Pipeline for Incident Detection*

Before applying the Machine Learning and Deep Learning algorithms explained in detail in Chapters 5 and 6, the data stemming from the data sources need to be collected and prepared accordingly (data processing and cleaning, outliers removal, feature extraction and engineering, feature scaling and selection) and after the application of the respective algorithm, the model needs to be evaluated and fine-tuned before deployment. The different steps of the pipeline are illustrated in Figure 8-2, based on the pipeline proposed in (ENISA - European Union Agency for Cybersecurity, 2020)) and are briefly described below.

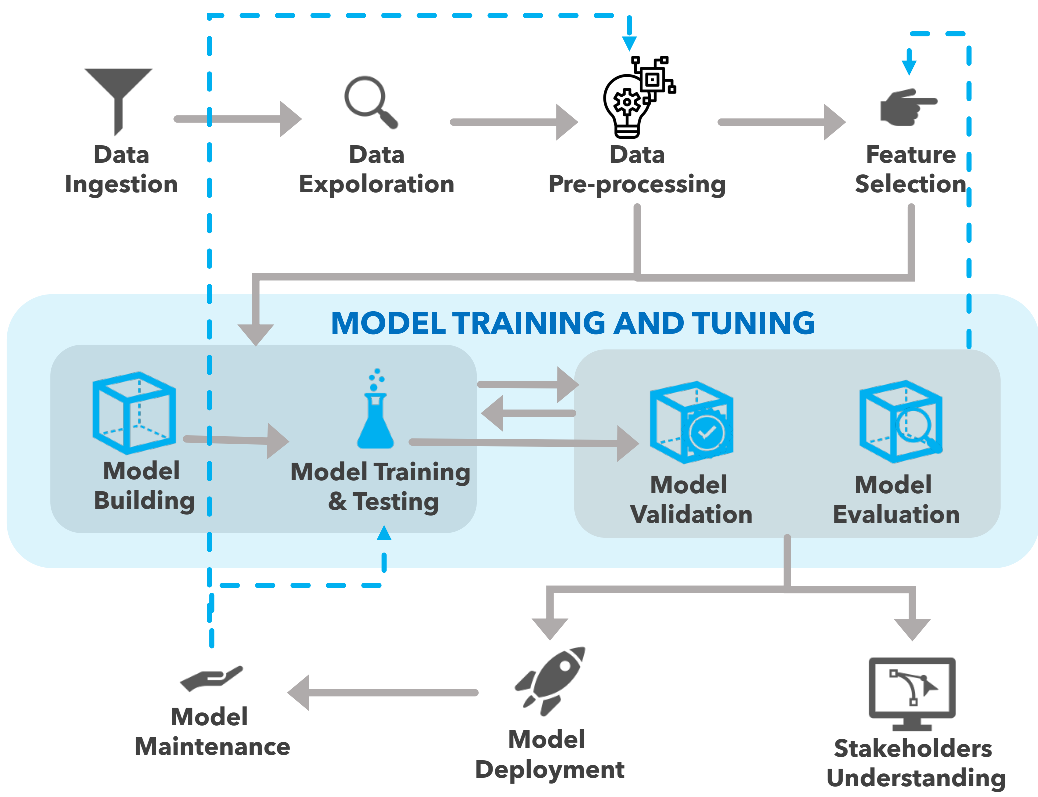



**Figure 8-2: Data processing pipeline for data-driven Incident Detection (based on the pipeline proposed in (ENISA - European Union Agency for Cybersecurity, 2020)).**

#### 8.1.1.1 Data ingestion

Firstly, the data needs to be obtained from multiple sources in order to compose multi-dimensional data points, called vectors, for immediate use or for storage in order to be accessed and used later. Data Ingestion lies at the basis of any AI application. Data can be ingested directly from its sources in a real-time fashion, a continuous way also known as streaming, or by importing data batches, where data is imported periodically in large macro-batches or in small micro-batches.

#### 8.1.1.2 Data exploration

At this stage, Data Exploration, insights start to be taken from the ingested data. While it may be skipped in some applications where data is well understood, it is usually a very time-consuming phase of the methodology's life cycle. At this stage, it is important to understand the type of data and basic characteristics of the data that were collected.

In our case, most of the data which are collected are numerical, for instance the measurements of loop detectors, or categorical, such as the types of incidents from the incident reports.

#### 8.1.1.3 Data pre-processing

The first step of the Machine Learning pipeline is the data pre-processing stage. In this stage, we employ techniques to cleanse, integrate and transform the data. This process aims at improving the data quality, which in a later stage will improve performance and efficiency of the overall AI system. Specifically, the term data cleaning designates techniques to correct inconsistencies, remove noise and eliminate faulty measurements. Moreover, in our case, in this stage, the reliability of the loop detector measurements of the traffic characteristics are calculated.

#### 8.1.1.4 Feature Selection

Feature Selection (in general feature engineering) is the stage where the number of components or features (also called dimensions) composing each data vector is reduced, by identifying the components that are believed to be the most meaningful for the AI model. The result is a reduced dataset, as each data vector has fewer components than before. Besides the computational cost reduction, feature selection can bring more accurate models. Additionally, models built on top of lower dimensional data are more understandable and explainable.

#### 8.1.1.5 Model building

This stage performs the building of the best AI model or algorithm for analyzing the data. The three commonly identified major categories are supervised learning, unsupervised learning and reinforcement learning models.

Supervised techniques deal with labelled data: the AI model is used to learn the mapping between input examples and the target outputs. Some commonly selected algorithms are Support Vector Machines, and Neural Networks. Unsupervised techniques use unlabeled training data to describe and extract relations from it, either with the aim of organizing it into clusters, highlight association between data input space, summarize the distribution of data, and reduce data dimensionality. Reinforcement learning maps situations with actions, by learning behaviors that will maximize a desired reward function.

For the incident detection task, as already presented, both Supervised and Unsupervised approaches are suitable and thus could fit into our framework, since we do have at our disposal labels for the respective dataset, however these may be subject to errors, in addition to considering the imbalance in the classes (the normal conditions are many more in comparison to incident occurrences).

It is important to remark that model selection (namely choosing the model adapted to the data) may trigger further transformation of the input data, as different AI models require different numerical encodings of the input data vectors. Generally speaking, selecting a model also includes choosing its training strategy. In the context of supervised learning for example, training involves computing (a learning function of) the difference between the model's output when it receives each training set data item as input, and its label. This result is used to modify the model to decrease the difference. Many training algorithms for error minimization are available, most of them based on gradient descent. Training algorithms have their own hyperparameters, including the function to be used to compute the model error (e.g. mean squared error), and the batch size, i.e. the number of labelled samples to be fed to the model to accumulate a value of the error to be used for adapting the model itself.

#### 8.1.1.6 Model Training

Having selected an AI model, the training phase of the AI system commences. In the context of supervised learning, the selected ML model must go through a training phase, where internal model parameters like weights and bias are learned from the data. This allows the model to gain understanding over the data being used and thus become more capable in analyzing them. Again, training involves computing (a function of) the difference between the model's output when it receives each training set data item *D* as input, and *D*'s label. This result is used to modify the model in order to decrease the difference between inferred result and the desired result and thus progressively leads to more accurate, expected results.

The training phase will feed the ML model with batches of input vectors and will use the selected learning function to adapt the model's internal parameters (weights and bias) based on a measure (e.g. linear, quadratic, log loss) of the difference between the model's output and the labels. Often, the available data set is partitioned at this stage into a training set, used for setting the model's parameters, and a test set, where evaluation criteria (e.g. error rate, accuracy, recall, precision) are only recorded in order to assess the model's performance outside the training set. Cross-Validation schemes randomly partition multiple times a data set into a training and a test portion of fixed sizes (e.g. 80% and 20% of the available data) and then repeat training and validation phases on each partition. For our case, we deem that the most suitable approach is the Time Series Split cross-validation which sequentially splits the data into training and testing sets, ensuring that the validation set always comes after the training set in time. This is essential since it helps better evaluate time-dependent models by respecting the temporal order of observations, which is crucial for maintaining the integrity of time series analysis.

For the application of AutoML, the model training process has been streamlined by automating the selection of optimal algorithms and tuning hyperparameters, thereby expediting the development of highly accurate predictive models with minimal manual intervention. This approach allows for the efficient handling of complex datasets and accelerates the deployment of tailored models that can adapt to the dynamic nature of traffic patterns and incident occurrences.

#### 8.1.1.7 Model Validation and Evaluation

After having trained the model, this needs to be validated and evaluated. The process of maximizing a model's performance without overfitting or creating too high of a variance is referred to as model tuning. In machine learning, this is accomplished by selecting appropriate “hyper-parameters”.

Certain parameters define high level concepts about the model, such as their learning function or modality, and cannot be learned from input data. These special parameters, called hyper-parameters, need to be setup manually, although they can under certain circumstances be tuned automatically by searching the model parameters’ space. This search, called hyper-parameter optimization, is often performed using classic optimization techniques like Grid Search, but Random Search and Bayesian optimization can also be used. It is important to remark that this stage uses a special data set (often called validation set), distinct from the training and test sets used in the previous stages. In our case, we have selected the Grid Search optimization for selecting the hyperparameters of our models and based the performance evaluation on the measures presented in the following section.

##### *8.1.1.7.1 Performance evaluation measures*

To evaluate automatic incident detection algorithms, quantitative measures are typically used. Many different measures have been used in the literature, including precision, recall or detection rate, f1-score and false alert rate among others. Many different definitions of these measures exist, but below we present the most commonly stated. Some of these definitions are also used later in this dissertation to evaluate the developed models.

These metrics can be defined and calculated as follows (Simeone, 2018):

The precision of an automatic incident detection algorithm is the ratio of correctly predicted positive observations (incidents) to the total predicted positive observations (incidents).

$$Precision = \frac{Number\ of\ correctly\ detected\ incidents}{Total\ number\ of\ samples\ predicted\ as\ incidents}$$

The recall is the ratio of correctly predicted positive observations to all observations in the actual class. Recall is also commonly referred to as the detection rate. The Recall metric measures the model's ability to accurately identify all positive cases. A model will be judged as correctly detecting an incident if an alert was raised at any point during an incident.

$$Recall = \frac{Number\ of\ correctly\ detected\ incidents}{Total\ number\ of\ actual\ incidents\ in\ the\ dataset}$$

F1 Score is the weighted average of precision and recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy for instance, but F1 is usually more useful than accuracy, especially if there is an uneven class distribution.

$$F1\ Score = \frac{2 \times (precision \times recall)}{(precision + recall)}$$

These metrics have been widely adopted in the field due to their effectiveness in assessing algorithm performance (Zhou, Gandomi, Chen, & Holzinger, 2021).

Two other widely used metrics are the false alert rate and the mean time to detect.

The false alert rate is the percentage of the number of messages for which an alert was raised but no incident was occurring, to the total number of messages for which no incident was occurring (i.e. false positive rate). It should be noted that here, a message is used as a term to represent a collection of traffic metrics that cover a particular time period at a detector (or detection location).

$$False\,alert\,rate(FAR) = 100 \times \frac{Number\,of\,messages\,where\,an\,alert\,was\,raised\,falsely}{Total\,number\,of\,messages\,where\,an\,incident\,did\,not\,occur}$$

The Mean time to detect is the mean time taken (in minutes) to raise the alert for a correctly detected incident, over a given time period and area.

$$Mean\,time\,to\,detect(MTTD) = \frac{1}{n}\sum_{i=1}^{n}(A_i - O_i)$$

where n is the number of verified incidents, $A_i$ is the start time of an IDA's alert being raised, and $O_i$ is the start time of the corresponding incident.

For regression analysis and traffic forecasting, the following measures are widely used in literature (Plevris, Solorzano, Bakas, & Ben Seghier, 2022):

The Mean Squared Error (MSE) is a popular regression-related metric having to do with the average squared error between the predicted and actual values. It takes positive or zero values and is given by

$$MSE = \frac{1}{N}\sum\nolimits_{1}^{N}(p_i - r_i)^2$$

One major disadvantage of MSE is that it is not robust to outliers. In case a sample has an associated error way larger than the one of other samples, the square of the error will be even larger. This, paired to the fact that MSE calculates the average of errors, makes MSE prone to outliers.

The Root Mean Squared Error (RMSE) is also a frequently used measure of the differences between values (sample or population values) predicted by a model, or an estimator and the values observed. It is the square root of MSE. Unlike MSE, RMSE provides an error measure in the same unit as the target variable. It takes values in the range [0, +∞) and it is given by

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N}\sum_{1}^{N}(p_i - r_i)^2}$$

It should be noted that these metrics are closely linked, and an improvement in one may be transferable to degradation in others (Ghosh and Smith, 2014). For example, an IDA may be able to lower its sensitivity of raising alerts in order to reduce its false alert rate, but it would come at the cost of reducing its detection rate and increasing its mean time to detect.

In the literature, other stated measures of performance include:

- The feedback of traffic management centers (TMCs), including thoughts on IDAs' operational performance, usability, ease of implementation. Although this measure will be subjective, it is an important factor affecting the usefulness of IDAs in TMCs.
- The time needed to calibrate to a new location or urban setting. That is, to go from the raw data required, to detecting incidents in real-time.
- Once implemented, the frequency and time taken to re-calibrate the IDA to maintain performance.
- If trained on field data, the time span of the training data required.

Lastly, explainability can serve as a measure for performance by enabling traffic system operators and stakeholders to validate the reliability and soundness of the predictions made by the model. It ensures that the automated decisions made during critical incidents are transparent, allowing for accountability and enabling rapid, informed responses.

#### 8.1.1.8 Model Deployment

A Machine Learning model will bring knowledge to an organization only when its predictions become available to final users. Deployment is the process of taking a trained model and making it available to the users.

Currently, we have deployed the online mode of operation of the developed system. The insights garnered from the deployed system come through Orion Context Broker, which will be explained in next subsection (Chapter 8.2), facilitating a seamless flow of information and enhanced decision-making capabilities across the system.

#### 8.1.1.9 Model Maintenance

After deployment, AI models need to be continuously monitored and maintained to handle concept changes and potential concept drifts that may arise during their operation. A change of concept happens when the meaning of an input to the model (or of an output label) changes, e.g., due to modified regulations. A concept drift occurs when the change is not drastic but emerges slowly.

A popular strategy to handle model maintenance is window-based relearning, which relies on recent data points to build a ML model. Another useful technique for AI model maintenance is back testing. In most cases, the user organization knows what happened in the aftermath of the AI model adoption and can compare model prediction to reality.

For our case, the way we have chosen to handle this step in the process is for the selected model will be monitored and maintained periodically, in order to sustain the defined goals using techniques described in Chapter 7 whereas the evaluation results are available in Chapter 9.6.2.

## *8.2 Technical Architecture*

The technical architecture of the developed Information System AutoEventX is illustrated and presented below in Figure 8-3 and further explored and explained in detail in the following sections.

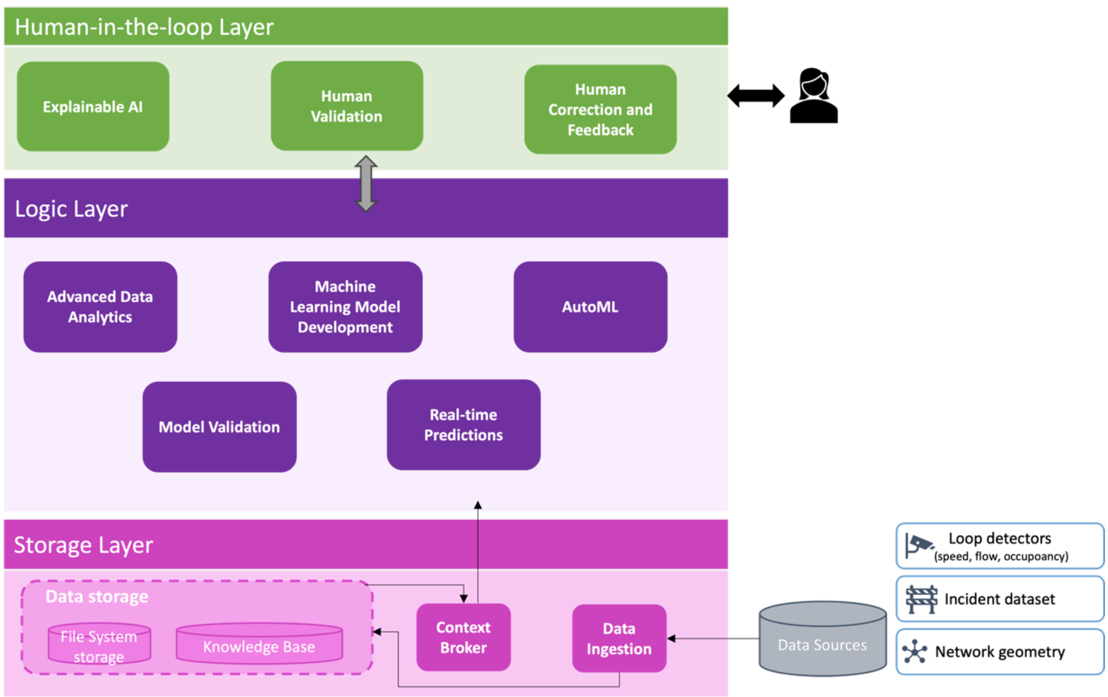



**Figure 8-3: Technical Architecture of developed Information System AutoEventX.**

### *8.2.1 Storage Layer*

The Storage Layer is responsible for collecting, storing, and managing the data, models, and analysis results required for incident detection. It ensures that the system has access to high-quality, relevant data and that all outputs are securely stored for future reference and analysis.

- **Data Sources**: The primary data sources include loop detectors, segment-level measurements, historical incident records, and network topology. These sources provide continuous, real-time data as well as historical data for training and validation purposes.

- **Data Ingestion**: This component involves the extraction, transformation, and loading (ETL) of data from various sources into a centralized data repository. It is crucial to ensure that data is consistently and accurately collected.

- **Data Storage**: The collected data is stored in scalable and secure databases. It consists of one non-relational NoSQL database, specifically document-oriented database, the Knowledge Base, in addition to an extended file system, which includes raw and processed parquet and json files, with the addition of the enhanced dataset after the inclusion of the human in the process. Moreover, the models are stored in this system, a fact which ensures that the most current and effective models are always available for use. The results of data analyses, including predictive models, feature importance scores, and other relevant outputs, are stored for future reference and further analysis. This helps in maintaining a comprehensive record of all analytical activities and outcomes.

In Figure 8-4, the schema of the data processing is represented. The data stemming from the aforementioned sources end up in a data lake to go through a process of fusion and harmonization (when required), prior to be stored in the corresponding database in the Data Storage layer.

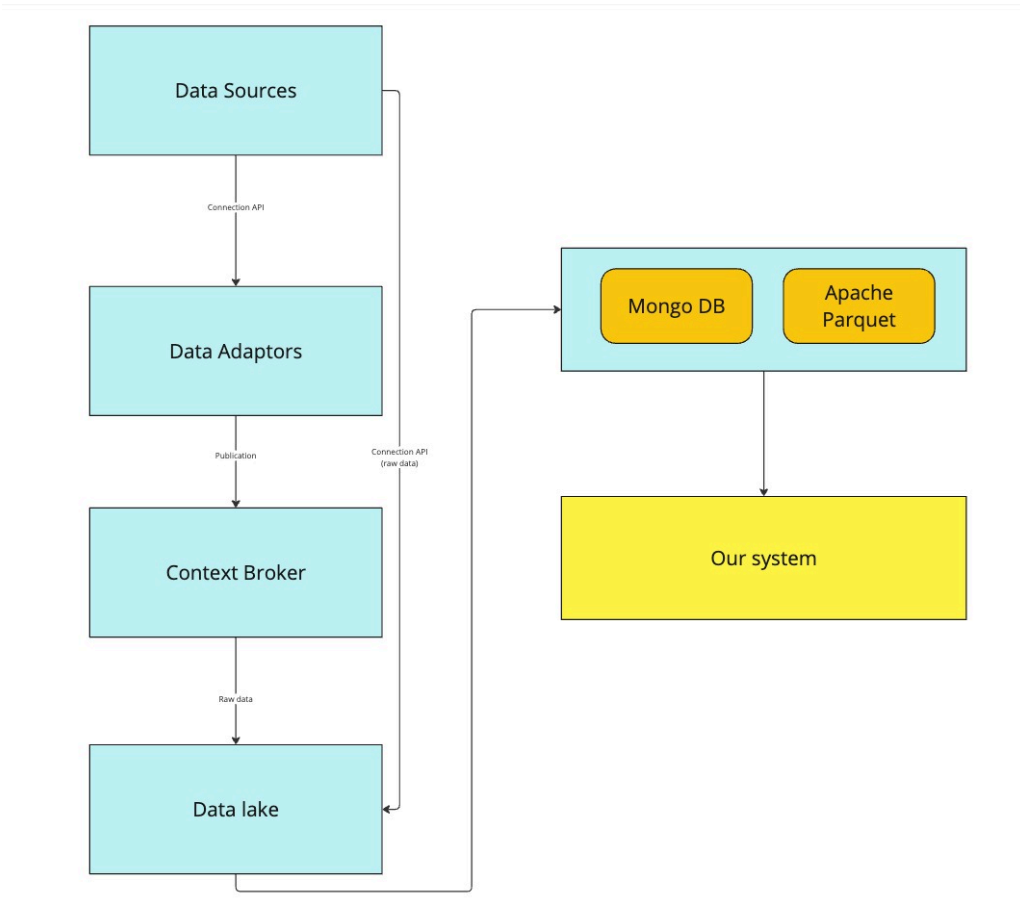



**Figure 8-4: Data management schema.**

### 8.2.1.1 Data Sources

#### *8.2.1.1.1 Inductive Loop Detectors (ILD) Dataset*

Inductive Loop Detectors (ILDs) are a core technology as traffic data sources, known for their robustness, reliability, and cost-effectiveness. These fixed sensors are embedded within the roadway surface and operate on the principle of electromagnetic induction. When a vehicle passes over or stops on the loop, the inductance in the circuit changes, triggering a signal that is recorded and processed.

One of the primary advantages of ILDs is their high level of accuracy in detecting vehicle presence and counting. This accuracy stems from their direct interaction with the vehicle's metal mass, resulting in precise data with minimal error rates. Over decades of deployment, ILDs have proven to be exceptionally durable, requiring relatively low maintenance while providing continuous, real-time data. Their widespread adoption across the globe is a testament to their reliability and cost-efficiency.

Furthermore, ILDs are not just limited to basic vehicle detection; they can gather a comprehensive range of traffic parameters. These include vehicle speed, volume (the number of vehicles passing over a loop), occupancy (the percentage of time a loop is occupied by a vehicle), density (vehicles per unit length of the road), and queue length. Additionally, ILDs can be and have already been utilized to infer more complex traffic conditions, such as identifying congestion patterns, incident detection, and traffic flow dynamics.

The data collected by ILDs is crucial for traffic management systems, providing the foundation for real-time traffic monitoring, control strategies, and long-term transportation planning. Despite the emergence of newer technologies like video-based detection systems and radar, ILDs remain the most widely deployed and trusted traffic monitoring tool due to their long-established performance and cost advantages.

#### *8.2.1.1.2 Incident Dataset*

The Incident Dataset is a critical component in the study of traffic management and safety, especially in the context of developing and validating incident detection algorithms. This dataset typically consists of records of traffic incidents, such as accidents, breakdowns, road blockages, and other non-recurring, or recurring (e.g. recurring congestion) events that disrupt normal traffic flow.

The dataset is usually compiled from various sources, including traffic management centers, police reports, social media, and crowd-sourced platforms. It may contain detailed information on the type of incident, its location, time of occurrence, duration, severity, and the resulting impact on traffic conditions. Additionally, the dataset may include metadata such as weather conditions, road surface conditions, and visibility, all of which can influence the occurrence and detection of incidents. For our case studies, this dataset stems from the traffic operators and contains not only reports about time, location and duration of incidents, but also information about its severity, type and subtype, and more fields which are going to be described in each case study individually in Chapter 9.

The Incident Dataset is synchronized with data from traffic monitoring systems like ILDs, allowing for a comprehensive analysis of how traffic parameters change before, during, and after an incident. This synchronization is vital for the development of machine learning models and algorithms aimed at early incident detection, prediction, and mitigation strategies.

##### *8.2.1.1.3 Network Topology*

The Network Topology dataset provides a detailed representation of the physical and logical arrangement of the transportation network. It includes the layout of roads, intersections, interchanges, traffic control devices (e.g., signals and signs), and the location of traffic monitoring sensors, such as ILDs. The dataset typically features information on road hierarchy (e.g., highways, arterial roads, local streets), lane configurations, speed limits, and other critical infrastructure details.

A comprehensive Network Topology dataset is essential for accurate traffic modeling and simulation. It allows for the replication of real-world traffic conditions in a virtual environment, enabling researchers and traffic engineers to study the effects of various traffic management strategies, the impact of road modifications, and the behavior of traffic under different scenarios, including incidents. In the context of incident detection, Network Topology data is crucial for understanding how traffic flows through a given area and how it is likely to be affected by an incident. By integrating Network Topology with ILD and Incident datasets, it is possible to create advanced data-driven models that predict traffic disruptions.

#### 8.2.1.2 Data Ingestion

The data ingestion process is a critical phase, as it involves the collection and integration of various data sources necessary for effective incident detection. The data used in the work conducted is sourced both in real-time (online) and from historical archives (offline), from a combination of automated Python scripts and the collaboration with traffic management operators.

##### *8.2.1.2.1 Online Data Ingestion*

For real-time data ingestion, Python scripts have been developed to automate the process of collecting live traffic data from various sensors and external data sources. These scripts are designed to interface with Application Programming Interfaces (APIs) provided by traffic monitoring systems, enabling the continuous retrieval of data from the respective sensors.

The scripts are configured to handle data in a streaming fashion, ensuring that the system remains responsive to new data as it becomes available. They are equipped with error-handling mechanisms to manage potential issues, while the collected data includes a range of traffic parameters (vehicle speed, volume, occupancy), which are crucial for real-time incident detection and traffic analysis.

##### *8.2.1.2.2 Offline Data Ingestion*

Historical traffic data has been made available by traffic management operators. This dataset includes archived records from ILDs, incident reports, and other relevant traffic information collected over several years. The offline data is essential for training and validating the incident detection algorithms, as it provides a rich source of labeled examples of traffic incidents and their associated traffic patterns. To ingest this offline data, Python scripts have been developed to automate the extraction process. These scripts are tailored to handle various data formats provided by the operators, including CSV and parquet files in addition to JSON records.

Both online and offline data streams are integrated into a unified data management framework that allows for seamless access and analysis. This framework is designed to support continuous updates from online sources while maintaining the integrity of the historical dataset. The entire data ingestion pipeline is managed to ensure data quality and consistency, which are critical for the performance of the incident detection models. The integration process includes the synchronization of timestamps, alignment of data formats, and the resolution of any discrepancies between the online and offline data sources.

#### 8.2.1.3 Data Storage databases

Relational databases are structured according to a model consisting of different data tables interconnected by foreign key relationships. Consequently, to answer a query or insert a new entry in a relational database, many tables are traversed and combined to gather or generate the requested information. In contrast, document-oriented databases, which are a subclass of key-value databases, do not follow a strict data schema but use document formats like XML, JSON, YAML, etc., to store all necessary information about an object in a single document, which can have a different structure from other documents in the database.

Since relational databases require a predefined schema before the construction of the database, any schema changes after data insertion can lead to problems. Conversely, document-oriented databases overcome this limitation and support a dynamic schema. This capability is useful for large and diverse data applications where adding documents with different structures is required without modifying existing data or the application itself.

In summary, NoSQL databases offer the following advantages (Gupta, Gupta, & Mohania, 2012):

- Support for large-scale data
- High write performance
- Fast key-value access
- Flexible schema, flexible databases, and easy schema conversion
- Ease of use for developers
- Support for distributed systems

According to (Han, Haihong, Le, & Du, 2011), NoSQL systems are categorized into three types:

- **Key-value databases:** Each value corresponds to a key. These databases, with a very simple structure, provide much higher speed than relational databases and support massive storage with high concurrency. A representative example is Redis.
- **Column-oriented databases:** These databases organize data in tables without supporting table relationships. Data is stored by column, where each column serves as an index for the database. This reduces system I/O as only the necessary columns are traversed for each query. Additionally, these databases support simultaneous queries. An example is Cassandra.
- **Document-oriented databases:** These databases resemble key-value databases but with the difference that the value is a semantic object stored in XML or JSON format. These databases support secondary indexes on values, which are not supported by key-value databases. An example is MongoDB.

As described by (Han, Haihong, Le, & Du, 2011), modern large-scale data management applications require:

- High support for parallel data entries and retrievals with low latency
- Efficient large-scale data storage and support
- High availability and scalability
- Low operational and management costs

Under these conditions, relational databases exhibit low data write and retrieval speeds, limited capacity, and scalability difficulties. For these reasons, NoSQL databases facilitate large-scale data analytics, particularly for machine learning and reinforcement learning applications, providing increased scalability and high performance (Konstantinou, Angelou, Boumpouka, Tsoumakos, & Koziris, 2011).

For our case, the Knowledge Base is structured in a way that ensures that the data can be efficiently stored and retrieved by the other structural components of our system. Specifically, it contains information related to the task at hand, the identification of incidents. Even though it could be a relational database, to ensure consistency in the way data are accessed and retrieved using Orion Context Broker, we have selected to use a NoSQL Database, namely MongoDB. Below, we describe the primary entities and their relationships as depicted in the Entity-Relationship (ER) diagram shown in Figure 8-5:

1. **Site**: Represents a specific location with an intelligent transport system. Each site is uniquely identified by a siteId and includes attributes such as name, mapCenter, dateLocale, and displayName. This entity is crucial for categorizing and managing data related to various geographical areas.
2. **Organization**: This entity represents the different organizations that manage or interact with the transport system. It is identified by an organizationId and includes fields such as name, siteId, and authId. The siteId indicates the association of the organization with a particular site.
3. **Event**: Central to the incident prediction framework, the Event entity captures detailed information about specific incidents or occurrences. It is identified by

an eventId and includes attributes like name, description, siteId, location, eventType, expectedImpact, and probabilityOfOccurrence, among others. This entity records both the planned and unplanned events within the system, which are crucial for model training and prediction.

4. **EventStatusChange**: This entity tracks the status changes of an event over time. Identified by changeId, it logs each status update with attributes such as changeSequence, eventId, organizationId, status, changeTime, and reason. This allows for a detailed timeline of how an event evolves and is able to capture and track changes introduced by operators.

5. **EventAck**: The EventAck entity captures acknowledgments of events by various organizations. It includes an ackId, eventId, organizationId, ackTime, and any associated comments. This entity is essential for confirming that incidents have been acknowledged.

6. **User**: Represents the users interacting with the system. Each user is uniquely identified by a userId and includes details such as organizationId, siteId, and displayName. This entity is vital for managing access and actions within the system.


**Figure 8-5: Entity-Relationship (ER) diagram.**

These entities and their relationships form the foundation of the Knowledge Base, enabling the efficient storage, retrieval, and management of data necessary for incident prediction within intelligent transport systems.

#### *8.2.1.3.1 Data storage infrastructure*

The approach is to build a data storage system consisting of a data lake where the raw data coming from various use cases in its original data format are stored prior to be converted to the data format and stored in a centralized data storage repository, as described above in detail.

#### *8.2.1.3.2 Data base technologies*

The type of data to be handled in the different case studies is very diverse. Consequently, to handle them efficiently has to deal with a combination of data storage technologies, adapted to the characteristics and usage of this information.

The technologies chosen to cover the identified needs in terms of data storage are:

- MongoDB
- Apache Parquet

Prior to storing the data in one of the aforementioned data bases, a process for fusion and harmonization takes place in the data lake to convert the data formats generated by origin data sources into an appropriate data model, as well as other processes that will enable consistent and high-quality data sets to serve as input to advanced applications. The structured and non-structured data will be stored in Apache Parquet and MongoDB respectively. The characteristics of the different data sources and the rationale for the selection of the type of information to be stored are described below.

Apache Parquet is an open-source column-oriented data storage system, suitable to store structured data. The algorithms that use Apache Parquet allow to accommodate complex data structures by using and efficient column-wise compression that saves storage space while offers efficient queries and the availability of different encoding techniques for different columns. In our case, the structured data coming from time series, especially those requiring extensive analytic operations, together with some sets of shapefiles are stored in Apache Parquet.

MongoDB is an open-source non-relational database (NoSQL) data storage system oriented to documents. Documents are semi-structured data that can contain any type of information or shape. Internally, it stores the data in a Binary-JSON (BSON) structure and allows to index the information with primary and secondary indexes to perform searches. The maximum BSON document size is 16 MB (MongoDB v5.0) which must be kept in mind while defining the document content. MongoDB provides high availability and replication and is very suitable to be used in a distributed way, if needed. In the context of this research, we have identified as suitable to store mobility data coming from XML, JSON and GTFS formats. This selected database is configured to be used to store definition of networks and areas, shapefiles, stations, schedules, traffic and other sensors characteristics and traffic events (accidents, roadblocks, road works, etc.), both real-time and historical.

Table 8-1 is a summary of the distribution of data types in the different storage technologies:

**Table 8-1: Relation of data and databases in the developed Information System.**

| Data Type | Original format | Database |
|---|---|---|
| • Static/near static object characteristics/properties/states (sensor, vehicles, network)<br>• Small/medium dynamic datasets (measurements)<br>• Events | XML<br>JSON<br>GTFS | MongoDB |
| • Time series<br>• Shapefiles | XLS<br>CSV<br>Shapefile | Apache Parquet |

#### 8.2.1.4 General data management structure

Among its objectives, our framework aims to facilitate data integration from different heterogeneous sources in an automated and standardized way, while ensuring data quality, and at the same time, being able to manage large data streams efficiently. The management infrastructure provides storage for both static data and near real-time data with different formats and access types. The first one, provided by the context broker and a REST service storage, and the second one consisting of a file system storage keeping the data in JSON, JSON-LD, and Parquet formats to maintain the original data for further system feeding and database restoration if required. The data provided comes from an ETL (Extract-Transform-Load) process, as shown in Figure 8-6, a timed Data Collector process is executed for each data source, each process is in charge of access to the corresponding data source to Extract the data and perform the required Transformation to the required data format to Load the data in its corresponding file system storage location as well as to format the data in the required format so it can be sent via POST request to Orion-LD.

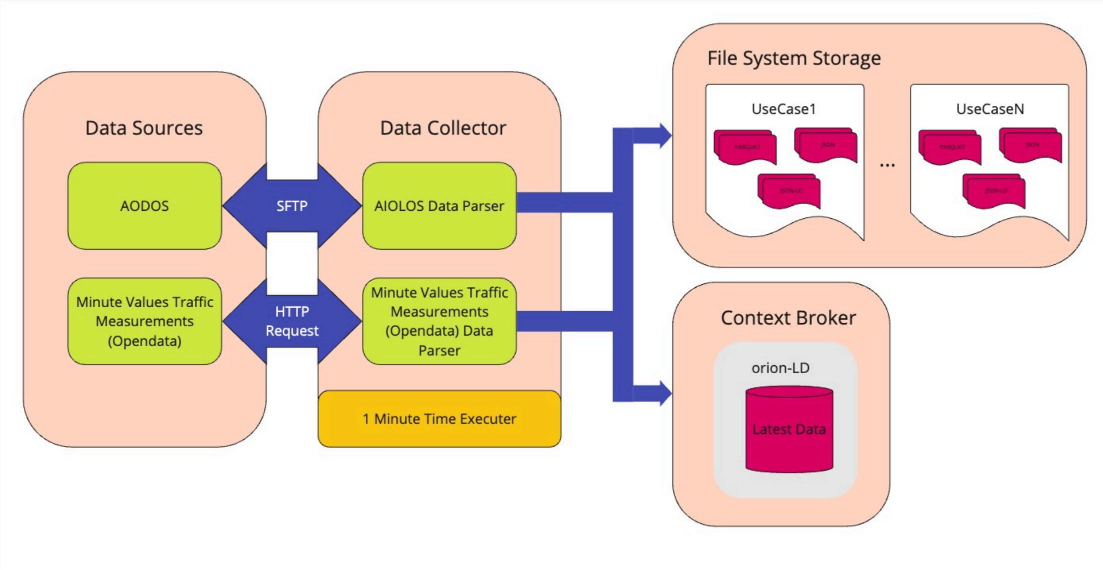



**Figure 8-6: ETL schema.**

### 8.2.1.5 Orion Context Broker for efficient data exchange

Data play an integral part in our methodology, that is why an analysis of the data available, its format and its suitable representation and storage to build applications has been performed. Nonetheless, this data must be exchanged between the subcomponents of our developed system, which in turn will generate new information based on basic data inputs. To make all of them interoperable, it is important to set up a common framework (information representation) and communication channels to enable the data generated by producers to reach data consumers. The component that will fulfill this function is the context broker, while the common framework is given by using NGSI-LD.

A Context Broker acquires contextual information from heterogeneous sources and merges it into a coherent model that is then shared with entities in a distributed ecosystem. The contextual information refers to the information that is produced, harvested or observed and that could be relevant for processing, analysis, and extraction of new knowledge. Each piece of information or context element has associated one or more triples that refer to the attributes of the context element and a defined value.

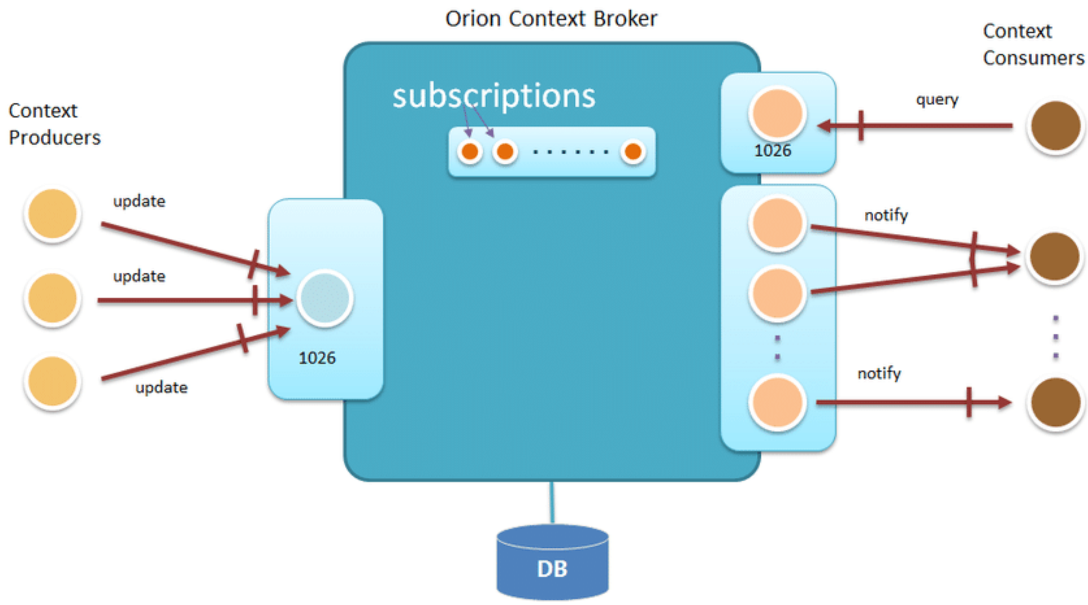



**Figure 8-7: Context Broker functioning schema. (Celesti, et al., 2019)**

The context broker selected to deal with the information sharing part is the Orion Context Broker (OCB). The OCB is a component developed by FIWARE that allows to manage, query and update context information. This allows to publish context information by some entities, called context producers, like sensors and make it available to other entities, called context consumers, which are interested in processing such information, as illustrated in Figure 8-7. This publication-subscription system allows decoupling data sources from other parts of the architecture. The communication is bidirectional, and a specific entity can be producer and consumer. The OCB acts as a server that includes an API based in the NGSI-LD (Next Generation Service Interface) model information, which allows to store actualized context information from the different sources, and solves queries based on this information. Eventually, a context consumer can take care of recording historical information in a separate database.

For our case, Orion, the chosen context broker, is responsible for managing the lifecycle of context information. To ensure this responsibility, the context broker provides an API which is useful for easing the data insertion. Once this data is received, the context broker stores it in a No-SQL database, ensuring the access to latest data received, so that it can be accessed through the same API as inserted, providing fast access to the newest data. The context broker offers a JSON-LD API with the necessary endpoints for creating, retrieving, updating, and deleting entities.

An important functionality offered by Orion is the capacity of creating subscriptions for receiving updates of the information in real time. Using this mechanism, a client can request the context broker to notify them on certain updates in the data. This is achieved using the "*subscribe*" operation. This operation allows the client to specify the notification channel. Moreover, the client can focus on specific data of interest by providing filters over the entity id, entity type, attribute, etc. Once subscribed, whenever a data provider updates an entity that matches the filters provided by the client in the subscription operation, the context broker will automatically notify the client of this event.

#### 8.2.1.6 Data format

Once the mechanism is set, we need to establish a common language to enable interoperability among components. The main elements that will enable that are described in the following subsections.

The Next Generation Service Interface Linked Data (NGSI-LD) is an information model and API used for an open and structured data exchange between the different stakeholders though a process of edit, query and subscription. NGSLI-LD has been standardized by the European Telecommunications Standardizations Institute (ETSI). The information model represented by NGSI-LD represents the context information as entities and their relations with other entities. The structure is acquired from the knowledge graph and the semantics described in the ontology of the system to study and defined formally with the Resource Description Framework (RDF).

To standardize and make available the data saved in the  respective storage to third parties when required, an ITS standard-based data model has been adopted, allowing the subsequent management of the heterogenous data sources for further processing. Here, the DATEX-II standard is considered for traffic related data. DATEX-II is the European standard for the exchange of traffic related data. It is a unified XML-based format modelled with UML (Unified Modelling Language) to allow data exchange between traffic management/control centres, traffic service providers, and road and traffic operators. It covers traffic and travel information such as:

- Traffic flow
- Traffic measures
- Roadworks
- Accidents
- Parking

The Orion Context Broker is a core component of the FIWARE platform, designed to manage context information at a large scale in IoT environments. It acts as a middleware that enables the integration and interoperability of various systems by providing a means to collect, manage, and disseminate context information. As an implementation of the NGSI-LD (Next Generation Service Interfaces for Linked Data) standard, Orion allows for the storage, retrieval, and subscription of context information in real-time, making it an essential tool for developing smart applications in various domains such as smart cities, industrial IoT, and more. (FIWARE)

One of the key features of Orion Context Broker is its ability to manage context data through a centralized system, which ensures data consistency and availability. It supports various data models and can integrate with multiple data sources, providing a unified view of the contextual data. This capability is particularly beneficial in scenarios where real-time data processing and decision-making are critical. For example, in a smart city environment, Orion can collect data from various sensors and systems (e.g., traffic lights, weather stations, public transportation) and provide real-time updates and notifications to city management systems, enhancing operational efficiency and improving citizen services (Gutiérrez, Martínez, & Sánchez, 2019).

Moreover, Orion's subscription and notification mechanism allows applications to subscribe to specific context changes and receive notifications when these changes occur. This feature supports proactive and reactive decision-making processes, which are crucial for dynamic and real-time applications. The scalability of Orion ensures that it can handle a large number of context updates per second, making it suitable for extensive IoT deployments (Wang & Chen, 2018). Additionally, its open-source nature and compliance with open standards facilitate customization and integration with other platforms and systems, promoting a collaborative and innovative development environment (Gyrard, Serrano, & Atemezing, 2017). In the data layer of many systems, including advanced IoT frameworks, Orion is used to ensure efficient data management, providing a backbone for handling contextual information (Smart Data Models).

### *8.2.2 Logic Layer*

The Logic Layer is the core computational layer where data analysis, model training, tuning, evaluation and validation, in addition to system's predictions occur. It encompasses the implementation of traditional and automated machine learning algorithms and the execution of advanced analytics for planned and unplanned incident prediction.

- **Advanced Data Analytics**: This includes time-series analysis, spatiotemporal analysis, and correlation analysis to uncover deeper insights and improve model accuracy. Tools like ARIMA for time-series forecasting and geospatial libraries like GeoPandas for spatiotemporal analysis are utilized.

- **Machine Learning Model Development**: This component involves the pre-processing, cleaning, selection, training, and validation of machine learning models. The key activities include:

    - **Data Pre-processing**: This process consists of preparing raw data for model development by cleaning the data (identifying and correcting errors or inconsistencies in the dataset), normalizing and scaling (transforming the data to a common scale without distorting differences in the ranges of values) and extracting features from the raw data.

  - **Algorithm Selection**: Choosing appropriate algorithms based on data characteristics.

  - **Model Training**: Training the models on the pre-processed data to learn patterns and make predictions. First, this process entails splitting the dataset into training and validation sets. The model is trained on the training set by iteratively adjusting parameters to minimize the prediction error. Then, adjusting the algorithm's hyperparameters to optimize model performance is critical. This process is called hyperparameter tuning. Techniques like grid search and random search are used to find the best combination of hyperparameters.

- **Model Validation**: The validation of models' performance using appropriate metrics such as precision, recall, F1-score, to ensure robustness and reliability, is of outmost importance in our framework and implementation. Using cross-validation techniques to evaluate the model's performance is essential as part of this step. Finally, performing error analysis needs to be included in this step. This involves analyzing the types of errors the model makes to understand its weaknesses., e.g. examining false positives and false negatives to identify patterns or conditions under which the model fails.

- **Automated Machine Learning (AutoML)**: The integration of AutoML libraries and tools aim to automate the end-to-end process of applying machine learning, from data pre-processing to model tuning and evaluation, making the system more efficient and scalable.

- **Real-time Predictions:** The system is able to provide predictions in real-time regarding identified incidents both unplanned and planned anywhere in the network which covers the sensors having been included in the analysis.

Python has been selected as a programming language to develop the proposed system, since it is a versatile and powerful programming language widely used in data science, machine learning, and artificial intelligence due to its simplicity, readability, and extensive library support. In the components we developed, Python was used as the primary language to handle various aspects of data processing, model development, and visualization.

Pandas, a powerful library for data manipulation and analysis, played a crucial role in handling and analyzing structured data. It provides data structures like DataFrames and Series, which are ideal for data manipulation tasks. Pandas library was extensively used for data cleaning, transformation, and exploration, enabling efficient manipulation of datasets through operations like filtering, grouping, and merging. This functionality was essential for preparing the data for subsequent machine learning tasks.

Scikit-learn is another key Python library used in our system for developing and training machine learning algorithms. It offers simple and efficient tools for data mining and analysis, built on top of NumPy, SciPy, and Matplotlib. Scikit-learn was utilized for implementing various machine learning algorithms, model evaluation metrics, and tools for model selection and validation. Its utilities for preprocessing data and feature engineering were essential in building robust models.

Keras, a high-level neural networks API written in Python, was integral to the design and implementation of deep learning models within the system. Keras operates on top of TensorFlow and is particularly valued for its user-friendly, modular, and extensible nature, allowing for quick prototyping of models and experimentation with different architectures. It provides a high-level abstraction that simplifies the process of building and training neural networks, making it unnecessary to deal with low-level details. TensorFlow, an open-source machine learning framework developed by Google, served as the backend engine for Keras and was used to perform the computationally intensive tasks required for training and deploying machine learning models. TensorFlow is known for its ability to handle large-scale models and offers a flexible architecture that can be deployed across various platforms.

Finally, Seaborn and Matplotlib, two essential libraries for data visualization in Python, were used to create a variety of plots. These visualizations were crucial for understanding data distribution, relationships between variables, and patterns that could inform feature selection and model tuning. Matplotlib, the foundation library for creating visualizations, provided a wide range of customizable plots. Seaborn, built on top of Matplotlib, offered a high-level interface for drawing attractive and informative statistical graphics, simplifying the creation of complex visualizations and making it easier to plot data directly from Pandas DataFrames. Lastly, for the autoML implementation, regarding the technologies used, we have used extensively the python libraries TPOT, which has already been thoroughly explained in Section 6.

### *8.2.3 Human-in-the-Loop Layer*

The Human-in-the-Loop Layer integrates human expertise into the system to enhance decision-making, ensure model accuracy, and build trust in the AI system.

- **Explainable AI**: Explainability tools provide insights into the model's decision-making process. This transparency helps stakeholders understand how predictions are made and ensures that the AI system's decisions are interpretable and justifiable.

- **Human Validation, Correction and Feedback**: Traffic management professionals review and correct the model's predictions. This feedback loop is essential for refining and improving the model over time. Human corrections help identify and rectify any errors in the AI predictions.

- **Integration with Traffic Management Systems**: The validated outputs could be integrated into existing traffic management systems, providing real-time incident alerts to traffic management personnel. This integration ensures timely and effective responses to detected incidents.

Regarding the technologies used for the implementation, we have used extensively the python libraries SHAP and LIME, which have already been thoroughly explained in Chapter 7.

For the overall system implementation, we utilized Flask and Docker to create a robust, scalable, and easily deployable environment for our system. These technologies played crucial roles in ensuring that the system is efficient, maintainable, and capable of handling various deployment scenarios.

Flask is a lightweight web framework for Python that was used to develop the APIs needed for our system. Flask is known for its simplicity and flexibility, making it an ideal choice for building web applications and RESTful APIs. By using Flask, we were able to create a server-side application that can handle HTTP requests, manage routes, and interact with the machine learning models and data processing components. Flask provides the necessary tools to build a web interface through which users can interact with the system, send data, and receive results.

Docker was employed to containerize the entire application, including all the dependencies of the utilized libraries. Docker simplifies the process of creating, deploying, and running applications by packaging them into containers. Each container includes everything needed to run the application, such as the code, runtime, libraries, and system tools. This ensures that the application behaves consistently across different environments.

The use of Docker aims to achieve the following benefits:

- **Consistency:** By containerizing the application, we ensured that it runs the same way in all environments, eliminating issues related to differences in software versions or system configurations.
- **Scalability:** Docker containers can be easily scaled up or down based on the system's needs. This flexibility is essential for handling varying loads and ensuring that the system remains responsive and performant.

- **Isolation:** Each container operates in its own isolated environment, reducing conflicts between different components or services and improving security.

- **Portability:** Docker containers can be deployed on any platform that supports Docker, making it straightforward to move the application across different servers or cloud services.

## 8.3 Modes of operation

### 8.3.1 Offline Mode of Operation

In Figure 8-8, the technical architecture of the offline operation of our developed system is presented.

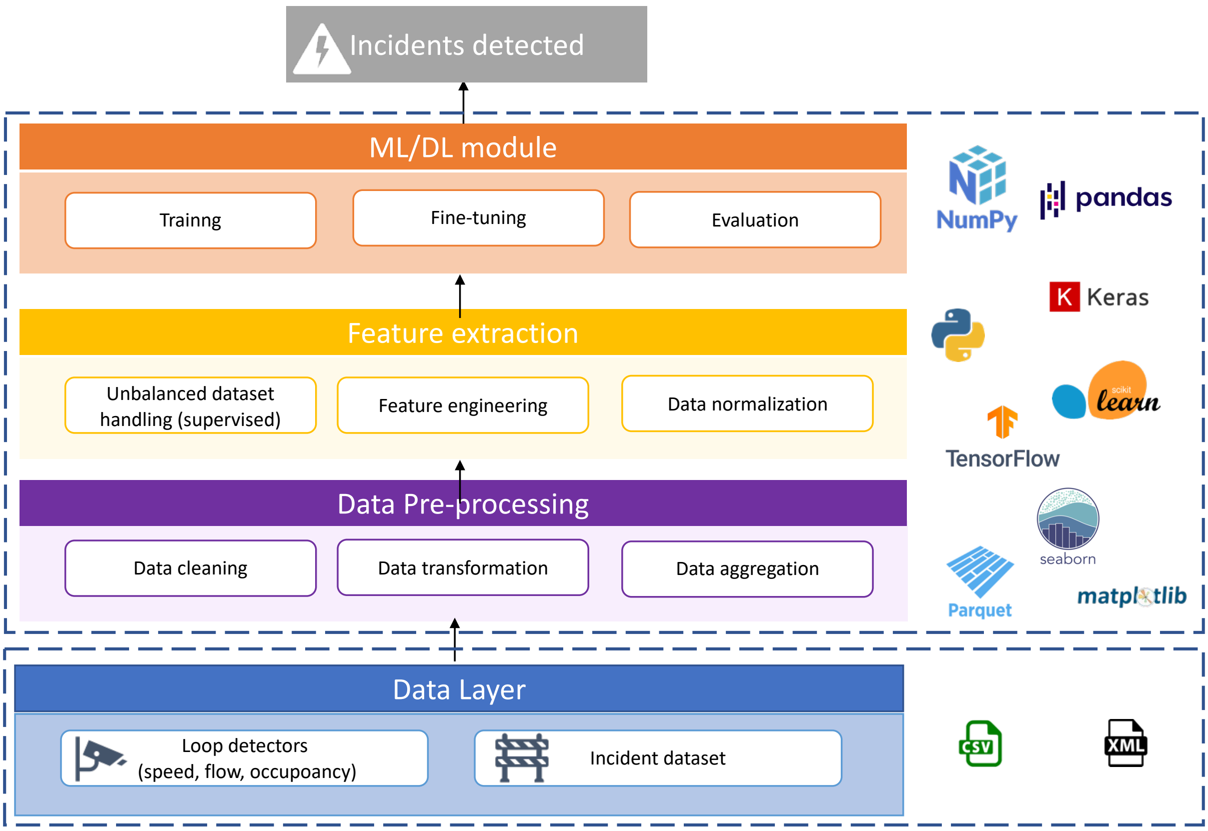



**Figure 8-8: Technical architecture of offline mode of operation.**

The Data Layer currently contains the loop detector (historical and real-time) measurements for speed, occupancy and flow in addition to the respective incident datasets and corresponding information about the network of each case. In the ML/DL module, we have implemented a suite of Machine Learning, Deep Learning algorithms and autoML algorithms for automatic incident detection. These include both Supervised and Unsupervised approaches.

### *8.3.2 Online Mode of Operation*

After having performed the training of the AI data-driven model in the offline mode as explained in the previous sections for non-recurring unplanned events and for recurrent congestion cases, our system is able to operate in real-time to raise alerts. Figure 8-9 displays the process flow of the online module of our system. As soon as new data becomes available, the online module of our system captures it. The data refresh rate can vary thus, the respective information needs to be collected, stored locally and then aggregated in specifically timed intervals to be fed in the pre-processing and data cleaning stage of the pipeline. The specific procedures for pre-processing remain consistent with those outlined in the offline mode of operation, maintaining uniformity in the approach to data preparation and cleaning. Then, the data are transformed in the required format to be fed in the step of model prediction. Should the entry contain anomalies (represented as "1"), then feedback is requested from operators, to confirm the identified incident. This human-in-the-loop concept is crucial, since it assists in creating a refined incident dataset and ensures that the system's performance could increase over time, given that it is retrained on this evolving dataset. It is worth mentioning that stakeholders can enhance the quality and accuracy of the reported incidents, by creating manual entries of identified incidents. Finally, in the case that the system has identified an anomaly in the data and labels it as incident, it then produces as output an entity of type "***Incident***" with the location and time attributes of the incident.

In order to enhance the system's detection capabilities over time, the feedback loop which we have implemented to compare model predictions with actual outcomes is key to our continued improvement. Detecting any discrepancies can be leveraged to optimize the model. Furthermore, implementing robust validation ny establishing a feedback loop for comparing model predictions with actual outcomes is crucial.

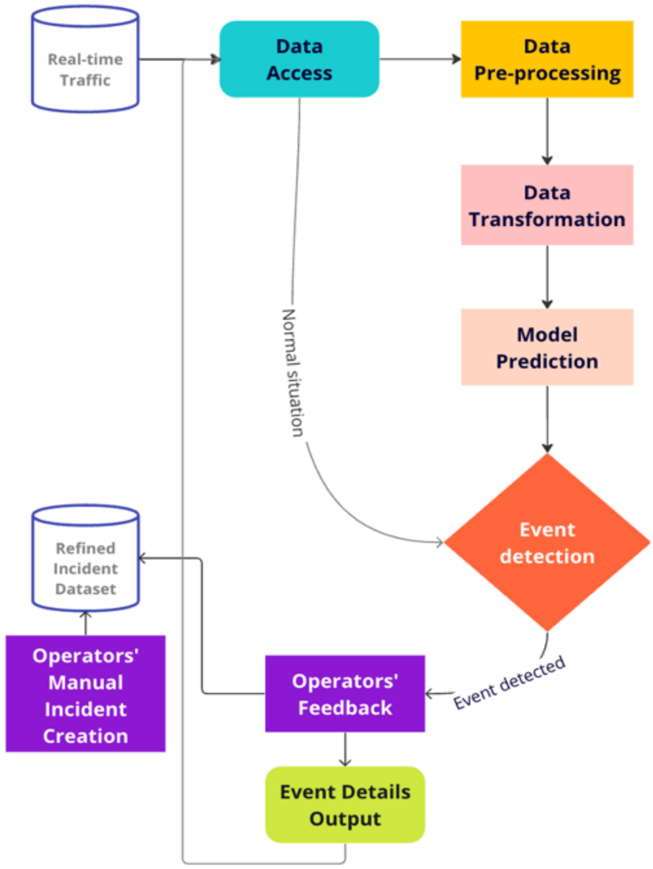



**Figure 8-9: Online mode of operation.**

## *8.4 Examples of system use*

This subchapter presents several examples from the use of the HITL (Human-In-The-Loop) traffic incident detection system developed as part of this dissertation. The following sections illustrate how the system identifies, explains, and refines traffic incidents, supported by screenshots from an external dashboard developed as part of the FRONTIER project.

### *8.4.1 Identification of Incident*

The system is designed to detect both planned and unplanned traffic incidents by analyzing real-time traffic data.

### 8.4.1.1 <u>Planned incident</u>

Below figures ( Figure 8-10 and Figure 8-11) illustrate a planned incident (recurring congestion) on the dashboard.


**Figure 8-10: Screenshot from dashboard depicting the identification of recurring congestion.**

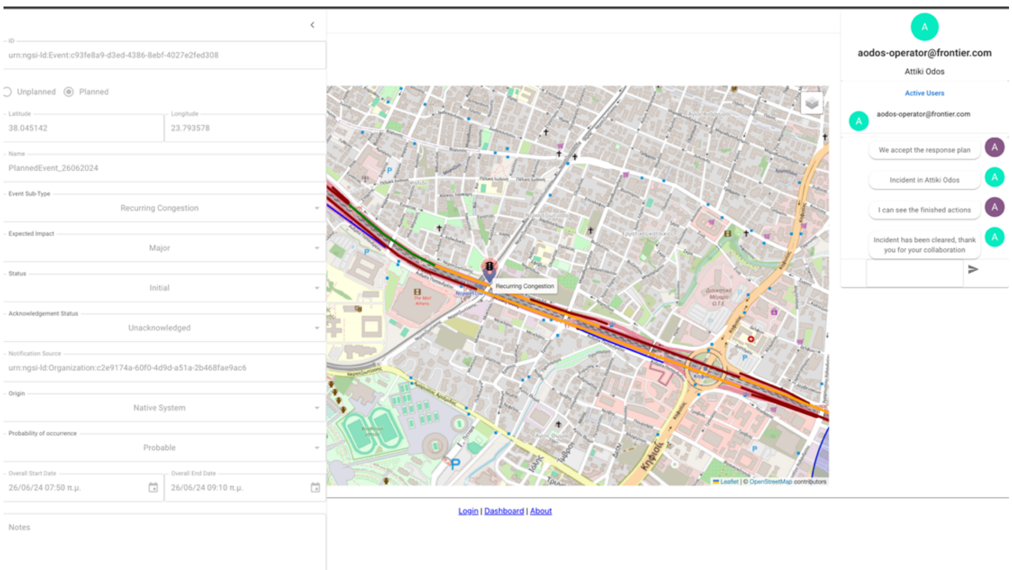



**Figure 8-11: Screenshot from dashboard depicting the details of detected incident (recurring congestion).**

#### 8.4.1.2 Unplanned incident

Figure 8-12, Figure 8-13 and Figure 8-14 show an unplanned traffic incident detected from the developed system AutoEventX on the dashboard.

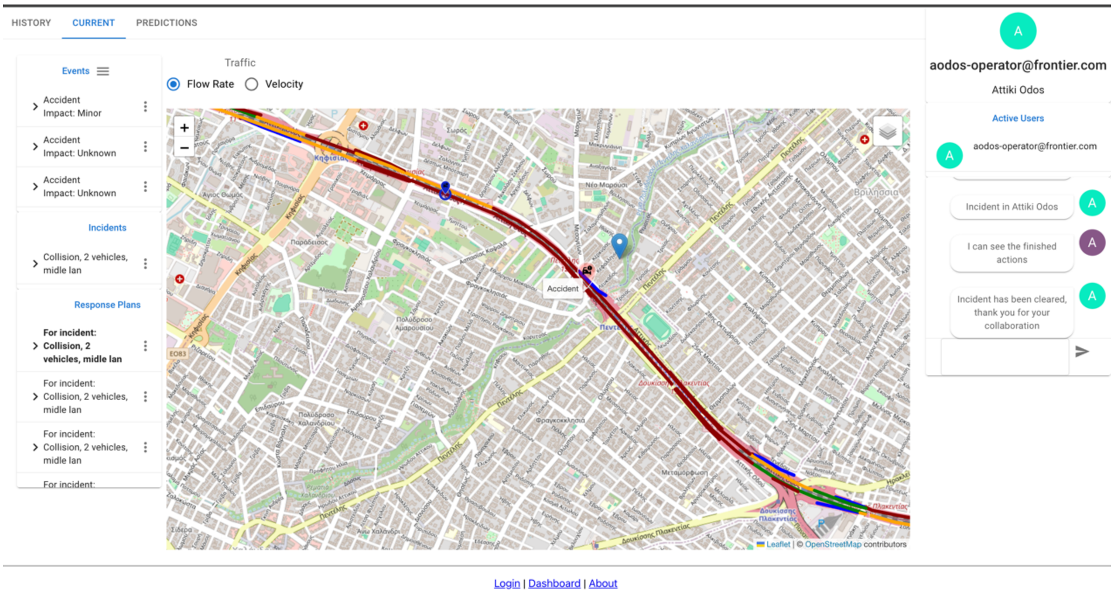



**Figure 8-12: Screenshot from dashboard depicting the identification of accident in a real-world case study.**

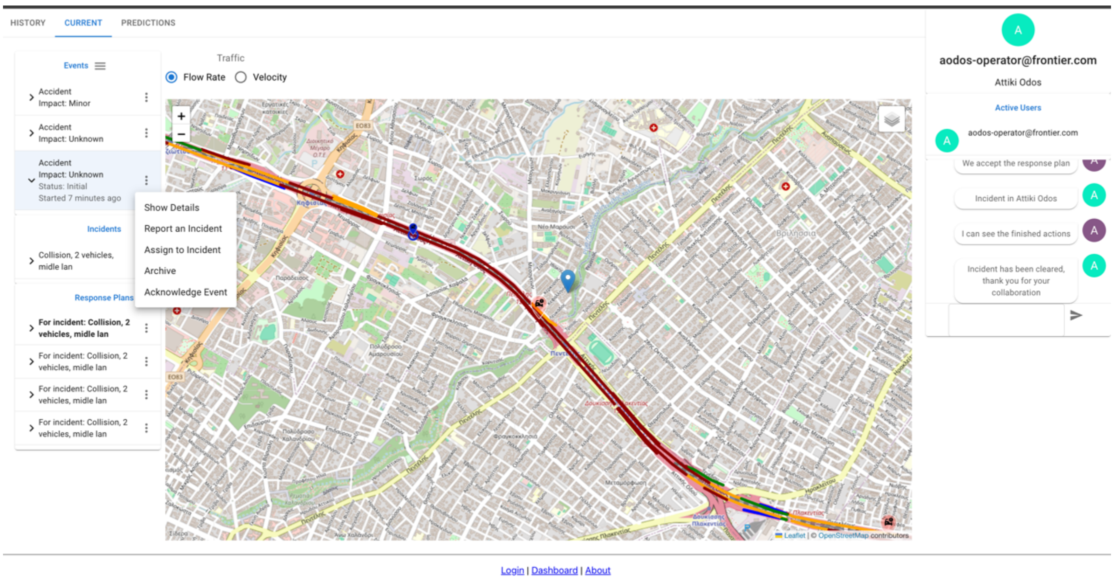



**Figure 8-13: Screenshot from dashboard depicting the panel and possibilities when an incident is detected.**

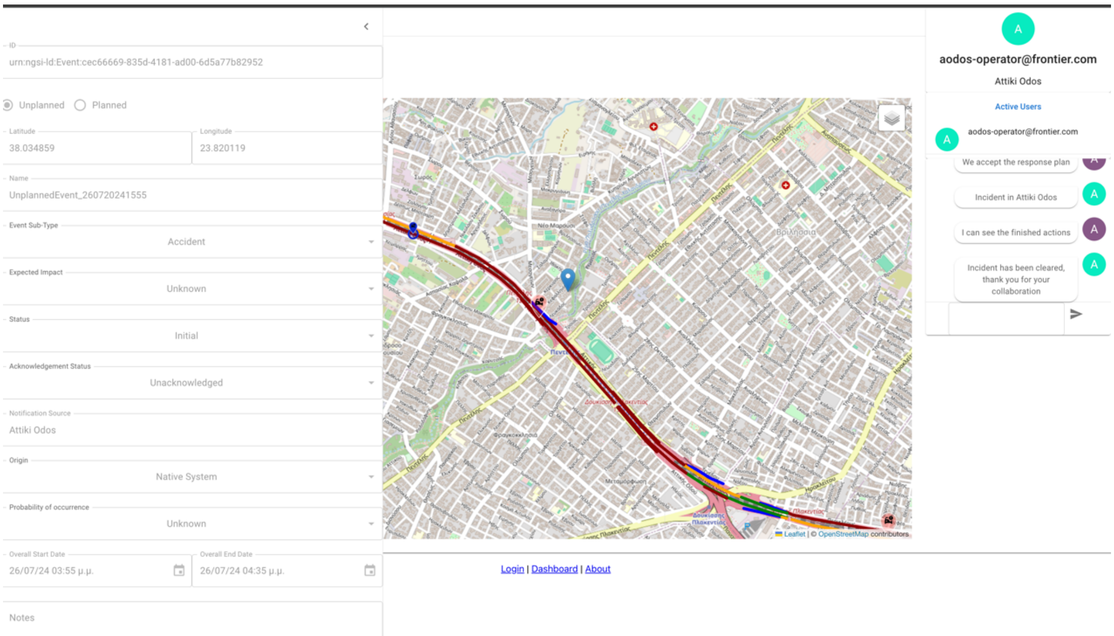



**Figure 8-14: Screenshot from dashboard depicting the details of detected incident (accident in real-world case study).**

### *8.4.2 Human Feedback*

Feedback from traffic management operators is essential for continuous improvement. The dashboard includes features for collecting and integrating user feedback. Operators can provide feedback directly through the dashboard, ensuring that their insights contribute to ongoing system enhancements (Gkioka, et al., 2024). This section details the feedback mechanisms available to operators.

#### 8.4.2.1 <u>Incident Validation</u>

Incident validation process as part of our system consists of the procedure by which operators provide important feedback regarding an automatically identified incident, towards refining the system performance and reliability. The objective of incident validation is dual: to confirm whether the incidents are correctly detected or to establish false detection that will need further refinement.

When the system detects an incident, operators verify its validity based on real-time data, contextual knowledge, and external information sources, such as phone calls from the impacted drivers. The operator will then validate the incident- which confirms that the incident indeed happened, and the impact is true- or reject it, where the incident would be classified as a false positive or wrong classification. In both cases, the operator gives the reason of rejection that is logged by the system for future analysis and improvement. Thus, incident validation reinforces the collaboration within human-in-the-loop systems, in the sense that the systematic integration of operators' feedback ensures better accuracy and higher operational efficiency of the deployed system.

In the context of our system, as soon as an event is identified automatically by the system, the operator is prompted to click on the right-hand side and select *"Acknowledge Event"*, as shown in Figure 8-15.

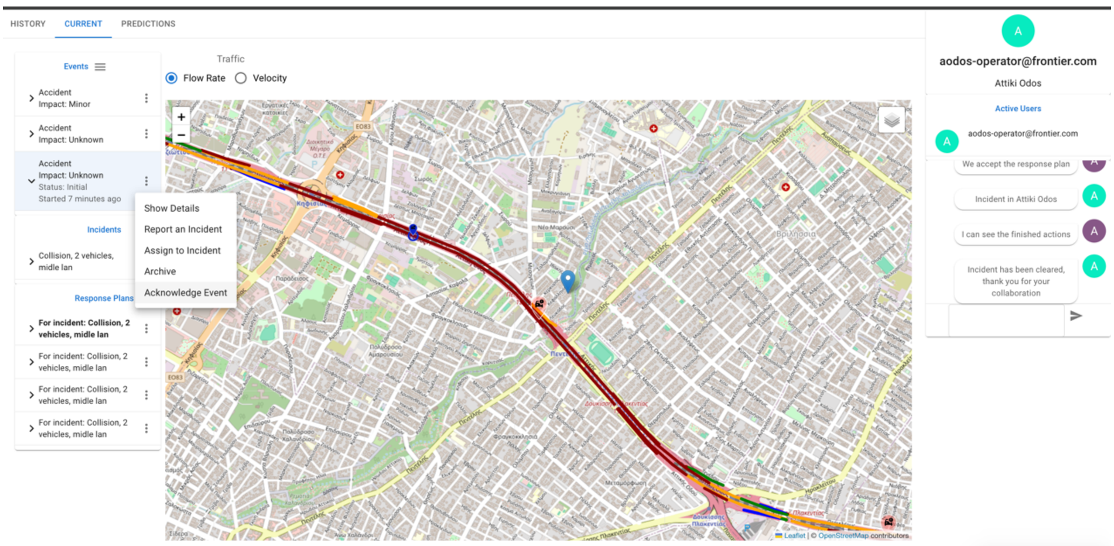



**Figure 8-15: The operator is prompted to validate the identified event through the dashboard.**

### *8.4.2.1.1 Verification*

Verification involves operators reviewing detected incidents and confirming their occurrence. When an incident is detected by the system, it is flagged for operator review. An example is described below:

- **Incident:** Traffic accident.
- **Action:** Operator reviews and confirms the incident.
- **Outcome:** The system records the incident as acknowledged.

This feedback loop helps the system learn from confirmed incidents, improving its detection performance over time.

An example of incident verification by an operator is illustrated in Figure 8-16.

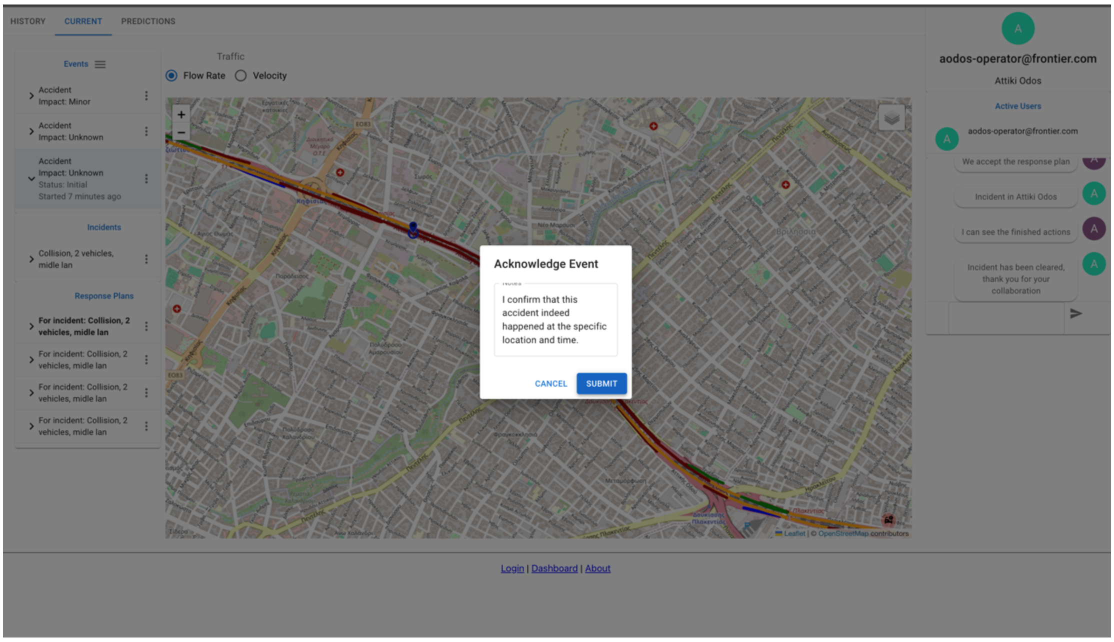



**Figure 8-16: The operator validates the identified incident through the dashboard.**

### *8.4.2.1.2 Rejection*

The system allows operators to reject detected incidents that are false positives or incorrectly identified. When an operator rejects an incident, they provide a reason for the rejection, which is recorded by the system for further analysis (automatically for instance using NLP or manually). An example follows for illustration purposes:

- **Incident:** System detects an unplanned incident, but the operator identifies it as a temporary slowdown and the traffic then gets back to normal conditions.
- **Action:** Operator rejects the flagged incident as false positive and notes the reason for rejection, if possible.
- **Outcome:** The system logs the rejection and the reason, with the aim of reducing similar false positives in the future.

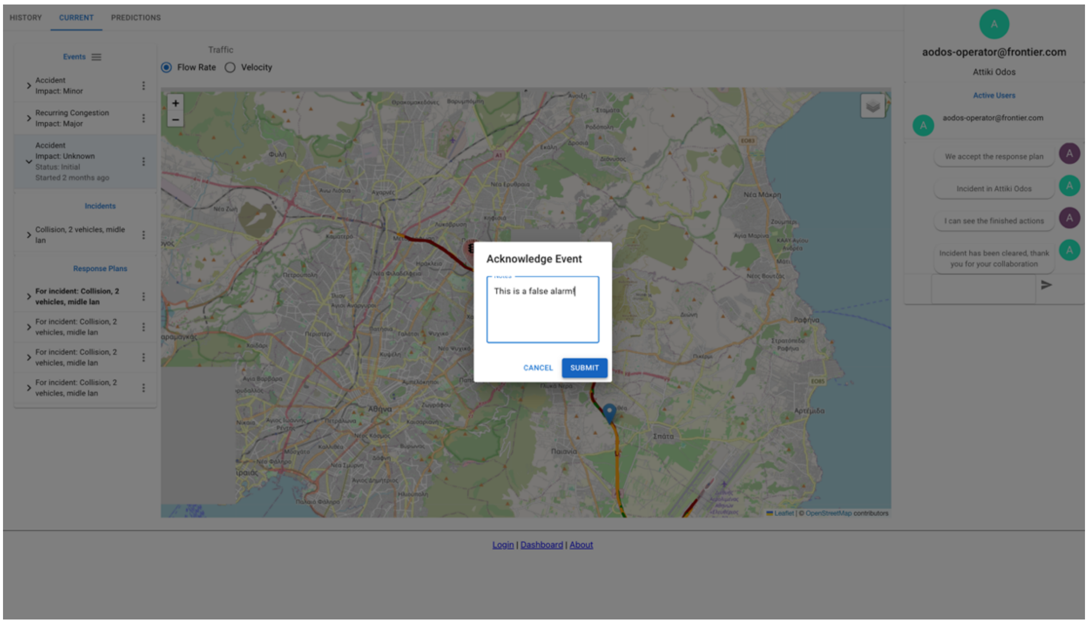



**Figure 8-17: The operator rejects the identified incident through the dashboard.**

This process is vital for refining the system's accuracy and reducing the occurrence of false alarms. However, the reason is logged only for informational purposes and is not used for retraining the models used as part of the system for incident detection.

An indicative example of an operator rejecting an incident is shown in Figure 8-17.

#### 8.4.2.2 Incident Insertion

In addition to validating detected incidents, operators can manually insert incidents that the system may have missed. This feature ensures that all relevant traffic events are accounted for, enhancing the comprehensiveness of the system's monitoring capabilities. An example follows for illustration purposes:

- **Incident:** Planned event in OAKA stadium.
- **Action:** Operator manually inserts the incident, including details such as location, duration, and expected impact.
- **Outcome:** The system updates its records and alerts drivers about the planned event and its potential impact. This allows for strategic traffic management.

By allowing manual incident insertion, the system benefits from human expertise and situational awareness, which can be critical in dynamic and complex traffic environments.

For this manual insertion, our system allows the operator to click on the map and click on the right-hand side to *"Create Planned Event"* or *"Create Unplanned Event*" according to the type of event identified by the operators, and then complete the fields as shown in Figure 8-18.

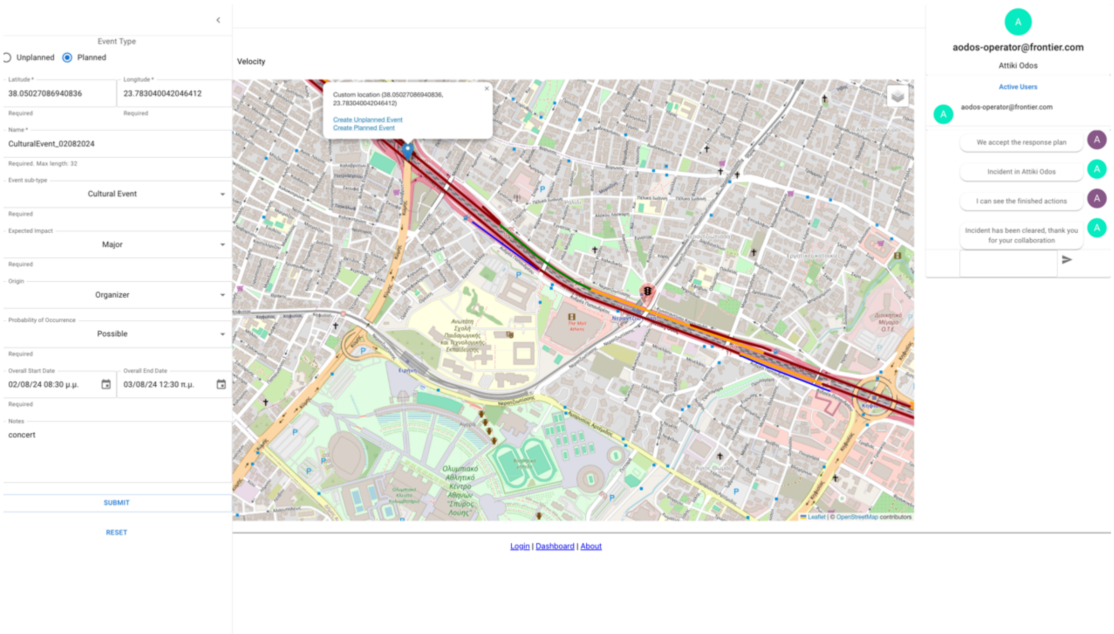


**Figure 8-18: The operator creates a planned or unplanned event with its details through the dashboard.**

# 9 Deployment and Evaluation in Real-world Case Studies

Chapter 9 discusses the results of the evaluation of the developed and deployed system in real-world case studies. In Athens, the system was evaluated in a dense urban corridor, addressing the unique challenges of traffic management in a historic metropolis. In Antwerp, the focus was on a critical route encompassing the city's port and major motorways. The deployment in these cities offered valuable insights into the adaptability and effectiveness of our AI-driven incident detection system across diverse urban contexts, demonstrating its potential for broader application in traffic management.

## *9.1 Case Study*

In the following section, we present real-life use cases from two distinct urban contexts, Athens, the capital of Greece, and Antwerp, a major city in Belgium, which validate the efficacy of our methodology within distinct urban environments. Athens provides a complex case with its dense urban network and the inherent challenges of a historic metropolis, while Antwerp offers a contrasting scenario with its strategic significance as a port city and its different network complexities. In Athens, we explore the application in a critical urban corridor, whereas in Antwerp, the focus shifts to a route connecting the city's port and major motorways. As part of this section, we detail the study area, in addition the datasets utilized in both cases.

### *9.1.1 Case Study I: Athens*

A corridor extending along 70 km and constituting the ring road of a metropolitan area connecting the airport to a populated suburb has been used as study area in our experiments. The road network model developed in the Aimsun Simulation Software

(Aimsun, 2023) is approximately 30 km long and involves 569 sections and 181 nodes and is depicted in Figure 9-1.

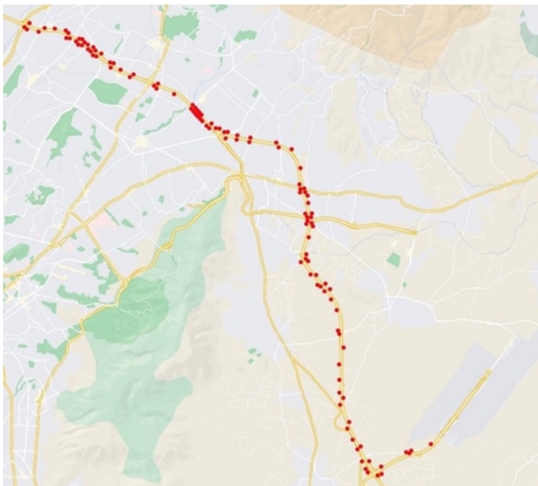


**Figure 9-1: The network of Athens study area.**

Loop detector data from 591 units were gathered from October 2020 to end of September 2021. Out of the total 591 detectors provided, 196 are regarded as reliable enough to be used as part of the experiments conducted. From the total amount of 26,331,086 readings provided (one every minute from the selected period), several filters were applied to remove detectors which were not in the station aggregation file, flow reliability outliers, flow-occupancy-speed mismatches, detectors with more than 50% not a number entries (NaNs), stuck values (constant readings across time), isolated values, and atypical profiles. Several types of imputation of missing/unreliable data were carried out on approximately 35% of the readings, namely: polynomial, time k-nearest neighbor (KNN), free-flow speed imputation, spatial KNN, PPCA-based imputation, and weekday-based imputation. Due to the low reliability scores of the loop detectors in occupancy and speed, the variable which was selected from the loop detector data to be used for the experiments was only the flow.

In addition to the Inductive Loop Detectors dataset, which comprises of the measurements of network-related attributes (i.e., speed, occupancy and flow), the labelled incidents dataset provided to us by the highway operator of our study area, plays a pivotal role in the experiments we conducted to validate our methodology. This dataset, comprising 34,652 incident occurrences in total and 34 feature columns, serves as a critical resource for evaluating the performance of our models, as it represents the ground truth against which our models will be assessed. By leveraging this dataset, we can measure the accuracy and effectiveness of our detection techniques, enabling us to make informed decisions and ensure the quality of the obtained predictions. The feature columns of this dataset include information regarding *'timestamp', 'source', 'start_time', 'end_time', 'direction', 'intersection', 'toll_station', 'branch', 'position_(pk)', 'type', 'subcategory', 'outcome', 'deaths', 'injured', 'queue_start_time', 'queue_end_time', 'queue_length_cars', 'queue_length_time', 'weather'* among others.

However, it is worth noting that certain inconsistencies were identified within the dataset, based on the conducted Exploratory Data Analysis. Specifically, incidents that had no discernible impact on traffic were still labeled as incidents. To ensure fairness in our experiments, a filtering process has been implemented to remove such instances, thus maintaining consistency in the type of loop detector input data used for analysis, based on the following:

- Notably, it was observed that two specific branches of the highway recorded the highest number of incidents, with 13,829 and 13,757 incidents respectively. Since the majority of the incidents occurred on the main branches of the highway, a decision was made to exclusively focus on those.
- Moreover, a filtering process was applied to include only specific incident types for the scope of our experiments. Specifically, the labelled incidents dataset exclusively encompasses incidents categorized as Traffic Congestion and Traffic Accident, as they are the primary focus of our investigation.
- Finally, the incidents were further filtered based on the observed queue length of cars. In collaboration with stakeholders, we obtained valuable feedback recommending a reduction in the threshold for queue length to 50 meters, as opposed to our initial proposal of 200 meters. This adjustment was made based on their expertise and supported by the understanding that queues of

200 meters are exceptionally uncommon in the specific highway, even in the event of an unplanned incident.

##### *9.1.1.1.1 Locations of the sensors*

For the city of Athens, a corridor of Attiki Odos (a modern motorway extending along 70 km and constituting the ring road of the greater metropolitan area of Athens) extending from the Athens airport to the suburb of Metamorfosi has been identified as the network which suit the identified needs for the evaluation of our framework. The road network model is approximately 30 km in size (it includes a section of the motorway) and involves 569 sections and 181 nodes as shown in Figure 9-1.

##### *9.1.1.1.2 Data collection*

For the data collection phase, we make a distinction between historical data and real-time data. Regarding the historical data, the end-user and data provider, Attikes Diadromes, has provided us with a folder containing raw data obtained from ILD from October 2020 until April 2021. For the real-time data collection, Attikes Diadromes, has provided access to an SFTP server which contains the raw data files of the last 24 hours. For this purpose, we have created and deployed a script which grabs the most recent files and stores them into a respective folder. The structure of the directory on the server where the raw data are stored follows the format: ***/year/month.*** Moreover, it gathers the content of the file (the raw data) and stores it directly in Orion Context Broker and the respective MongoDB in addition to parquet files, as described in detail in Chapter 8.2.1.

##### *9.1.1.1.3 Raw data characteristics*

The data is captured every minute from the ILDs in Attiki Odos, and each file contains the observations of speed, flow and occupancy stemming from each sensor. The format of the files containing the raw data is *xml*, and an example is shown below in Figure 9-2:

```xml
<oneMinuteData>
    <vdsData unitID="3944" timestamp="2021-05-01T00:00:00" speed="36" trafficFlow="60" occupancy="0" status="1"/>
    <vdsData unitID="3943" timestamp="2021-05-01T00:00:00" speed="0" trafficFlow="60" occupancy="0" status="1"/>
    <vdsData unitID="3945" timestamp="2021-05-01T00:00:00" speed="34" trafficFlow="60" occupancy="0" status="1"/>
    <vdsData unitID="3887" timestamp="2021-05-01T00:00:00" speed="0" trafficFlow="0" occupancy="0" status="1"/>
</oneMinuteData>
```


**Figure 9-2: Raw data in xml format.**

This real-time information about the tollway of Attiki Odos Motorway (Attica Tollway) in Athens, Greece provided by the Attica Tollway Operations Authority is illustrated and described in Table 9-1:

**Table 9-1: Data information for the 1Minute ILD Data.**

| XML Tag | DATEX II tag | Description |
|---|---|---|
| status | statusDescription | Sensor status |
| occupancy | occupancy | Road Occupancy |
| speed | averageVehicleSpeed | Speed |
| trafficFlow | vehicleFlowRate | Traffic flow |
| timestamp | timeValue | Datetime of the captured data |
| unitID | stationID | Id of the sensor |

A total of 591 detectors are registering flow, occupancy and speed in the original raw dataset. Preliminary analysis of this data shows that unitID 3944 is not providing consistent data at 60 seconds intervals, so it is flagged as a candidate to be discarded. However, as we will explain in more detail below, sensors show many inconsistencies in measurements which led us to contact the data supplier to provide us with a list of the most reliable sensors or a list of the unreliable ones.

A more detailed analysis of flow, occupancy and speed readings yield very low reliability scores for occupancy and speed. Reliability is estimated based on statistical analysis of the time-series, unknown values (NANs), zeros, negative values and outliers. Figure 9-3, Figure 9-4 and Figure 9-5 show heatmaps of flow, occupancy and speed variables, respectively, from 1-10-2020 to 30-09-2021. In these heatmaps, values in the color scale to the right of the figure depict the high (in green) to low (in red) quality of measurements from detectors as well as null readings (in black).

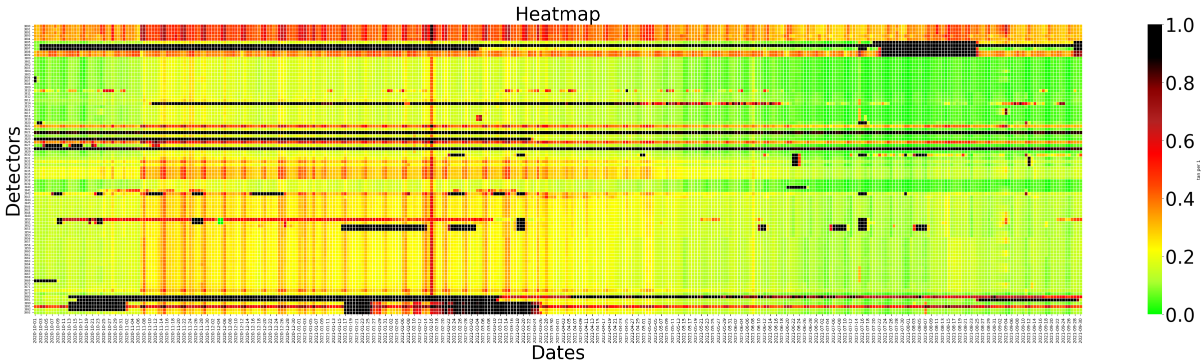



**Figure 9-3: Heatmap of flow raw data from 1-10-2020 to 30-09-2021.**

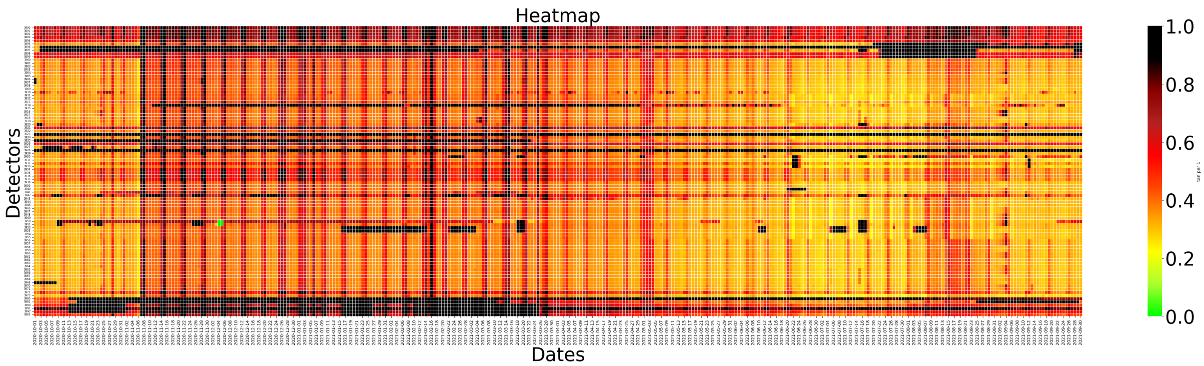



**Figure 9-4: Heatmap of occupancy raw data from 1-10-2020 to 30-09-2021.**

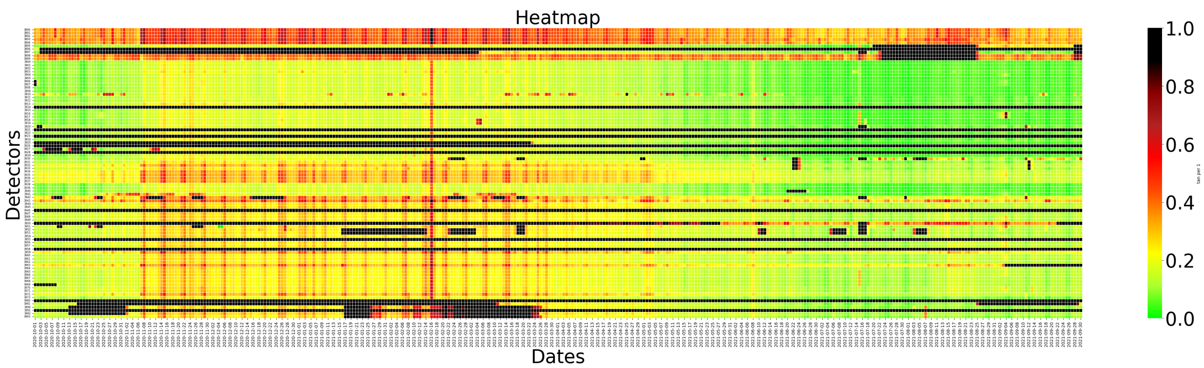



**Figure 9-5: Heatmap of speed raw data from 1-10-2020 to 30-09-2021.**

Results of this preliminary raw data analysis show an immediate need for data cleaning, which is detailed in the following subsection.

#### *9.1.1.1.4 Data cleaning and filtering*

To generate a high-quality dataset for learning, first the loop sensor data are pre-processed to be able to be fed in the Machine Learning and Deep Learning algorithms developed. Out of the total 591 detectors provided, only 196 are regarded as reliable enough. From the total amount of 26,331,086 readings provided (one every minute from Oct 2020 to April 2022), several filters were applied to remove:

- Detectors which were not in the station aggregation file
- Flow reliability outliers
- Flow-occupancy-speed mismatches
- Detectors with more than 50% NaN data
- Stuck values (constant readings across time)
- Isolated values
- Atypical profiles

Several types of imputation of missing/unreliable data were carried out on approximately 35% of the readings, namely:

- Polynomial
- Time k-nearest neighbor (KNN)
- Free-flow speed imputation
- Spatial KNN
- PPCA-based imputation
- Weekday-based imputation

Our methodology automatically discards low reliable sensors and data imputation involves lowering the reliability. Consequently, sensors with imputed data are not used for incident detection and it is recommended that any party utilizing the subsequent dataset either do likewise and discard low reliable data or experiment taking into consideration the implications this may have on their results.

#### *9.1.1.1.5 Cleaned data characteristics*

Figure 9-6 shows the final reliability of flow data (a sample visualization between 1-10-2020 and 30-09-2021 is depicted) after the cleaning, imputation and aggregation process has been done, resulting in 564 detectors (out of which 196 have got better reliability scores). Low reliability is depicted in black and high reliability in green.

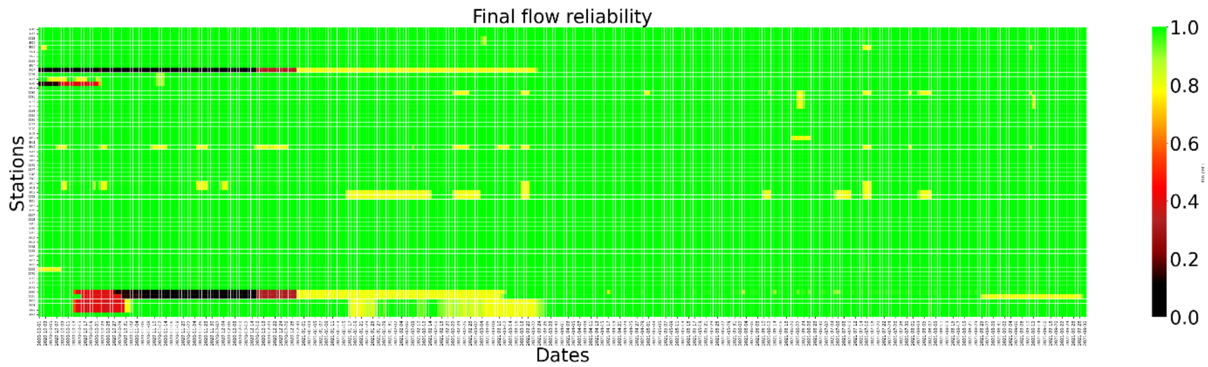



**Figure 9-6: Heatmap of flow cleaned data from 1-10-2020 to 30-09-2021.**

### *9.1.1.1.6 Data transformation*

Finally, the data are transformed and stored on the server. Each parquet file contains the monthly observations of one of the traffic characteristics (speed, occupancy, flow). Reading this file as a dataframe, this contains as an index the timestamp, in 5-minute intervals, and as columns the respective ILD ids, as identified by the raw data. The corresponding measurements are the values which fill the dataframe and characterize the ILD and timestamp. Moreover, there is a corresponding file which contains the reliability of each sensor for each observation in each timestamp where the later was captured. Finally, the data are also stored in Orion Context Broker and the respective MongoDB as described in the Chapter detailing the developed system.

Below you can find the representation of flow as illustrated in indicative dataframes (the speed and occupancy dataframes are similar regarding their representation):

```
columns               3001    3002    3003  3004  ...   3990  3991   3992   3993
index                                            ...
2020-10-01 00:05:00   0.00   282.0   216.0   0.0  ...  144.0  24.0  108.0  168.0
2020-10-01 00:10:00  21.00   324.0   342.0  24.0  ...  228.0  48.0  180.0  288.0
2020-10-01 00:15:00  30.00   270.0   276.0  48.0  ...  162.0  12.0  126.0  204.0
2020-10-01 00:20:00   5.25   408.0   432.0  12.0  ...  156.0  12.0  180.0  156.0
2020-10-01 00:25:00  11.25   336.0   318.0  36.0  ...   84.0  12.0   84.0   84.0
...                    ...     ...     ...   ...  ...    ...   ...    ...    ...
2022-06-06 23:40:00  24.00  1020.0   984.0  60.0  ...   72.0  24.0   72.0   96.0
2022-06-06 23:45:00  72.00   942.0   960.0  36.0  ...  108.0  24.0  108.0  120.0
2022-06-06 23:50:00  24.00   864.0   828.0  24.0  ...  168.0  12.0  132.0  156.0
2022-06-06 23:55:00  24.00  1032.0  1176.0  24.0  ...  156.0  12.0  144.0  156.0
2022-06-07 00:00:00  24.00  1098.0  1044.0  60.0  ...  132.0  12.0  132.0  132.0

[176832 rows x 564 columns]
```

**Figure 9-7: Dataframe including sensors' flow.**

```
columns                  3001  3002  3003  3004  ...  3990  3991  3992  3993
index                                            ...
2020-10-01 00:05:00  1.000000   1.0   1.0   1.0  ...   1.0   1.0   1.0   1.0
2020-10-01 00:10:00  0.024222   1.0   1.0   1.0  ...   1.0   1.0   1.0   1.0
2020-10-01 00:15:00  0.024222   1.0   1.0   1.0  ...   1.0   1.0   1.0   1.0
2020-10-01 00:20:00  0.024222   1.0   1.0   1.0  ...   1.0   1.0   1.0   1.0
2020-10-01 00:25:00  0.024222   1.0   1.0   1.0  ...   1.0   1.0   1.0   1.0
...                       ...   ...   ...   ...  ...   ...   ...   ...   ...
2022-06-06 23:40:00  1.000000   1.0   1.0   1.0  ...   1.0   1.0   1.0   1.0
2022-06-06 23:45:00  1.000000   1.0   1.0   1.0  ...   1.0   1.0   1.0   1.0
2022-06-06 23:50:00  1.000000   1.0   1.0   1.0  ...   1.0   1.0   1.0   1.0
2022-06-06 23:55:00  1.000000   1.0   1.0   1.0  ...   1.0   1.0   1.0   1.0
2022-06-07 00:00:00  1.000000   1.0   1.0   1.0  ...   1.0   1.0   1.0   1.0

[176832 rows x 564 columns]
```

**Figure 9-8: Dataframe including sensors' reliability of flow.**

To summarize, after this filtering process, the dataset used primarily originates from a closed-circuit television (CCTV) system, encompassing a total of 1,786 incident occurrences for the two main branches and more specifically 763 reported incidents for the same time period as the traffic measurements. Following data cleaning and filtering, it was necessary to transform the dataset into a format suitable for utilization by our algorithms, namely in 5-minute intervals where rows refer to timestamps and columns were the id of the loop sensors, and the values of the matrix were 1 if this location and time corresponds to an incident occurrence, or 0 otherwise.

### *9.1.2 Case Study II: Antwerp*

For the city of Antwerp, there is a multitude of inductive loop detectors available which provide one-minute readings regarding the network conditions, however, the area which has been deemed suitable to be used as a test bed of our incident detection task represents the corridor between the port of Antwerp and Eindhout including E313 motorway in both directions. This area includes 103 nodes in Direction 1 and 164 nodes in Direction 2 and in Figure 9-9 the locations of the sensors are depicted on a map.

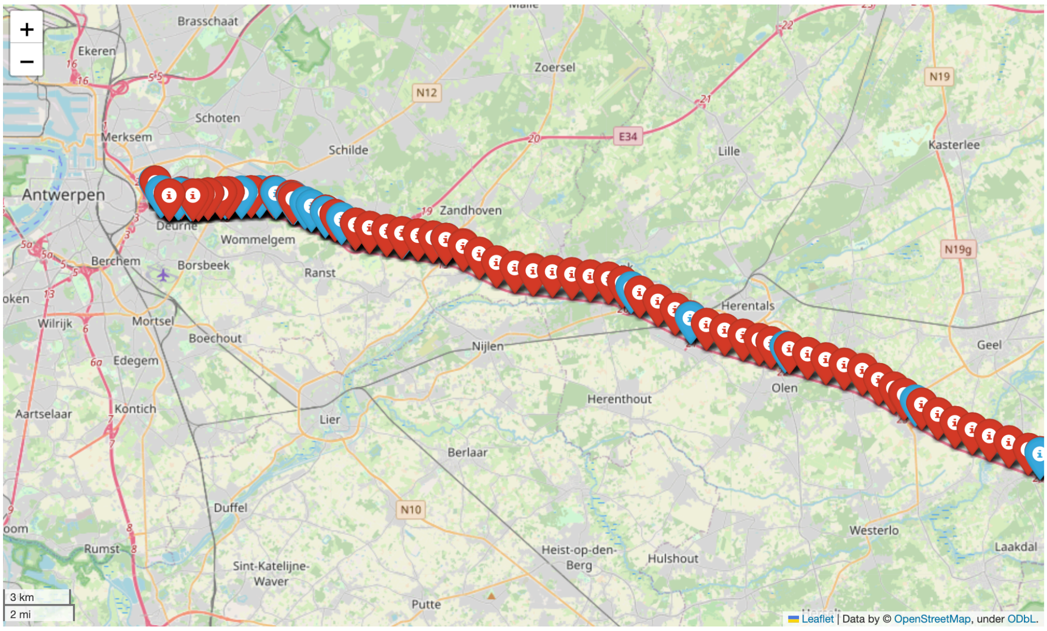


**Figure 9-9: Locations of loop detectors in Antwerp study area.**

The traffic data for E313 highway collected by loop detectors include the number of vehicles and the average speed, occupancy and flow, in addition to other traffic measurements, such as statusDescription, faultDescription and regularity, for 5 different vehicle classes/categories, aggregated per minute and the location of the measurement points. Loop detector data from 267 units was gathered from end of October 2022 to end of August 2023. The analysis of raw data yielded acceptable results in terms of quality of detectors' measurements, where one can observe some data gaps around May-June 2023 and a couple of missing days in Oct-Nov 2022. There are seven detectors that do not provide consistent readings over the whole period and those have been excluded from the analysis. In the pre-processing phase of our analysis, we employed a meticulous filtering and cleaning process to ensure the integrity and quality of the data. Initially, we identified and rectified any anomalies in the data, such as outliers or incomplete records. Subsequently, to streamline the dataset for more coherent analysis, we aggregated the different vehicle categories into a single, consolidated attribute. This means that we only kept the sum of vehicleFlowRate and the average of the speed, in addition to the supplementary attributes mentioned. This aggregation enables a more generalized assessment of traffic patterns while maintaining the robustness of the data. Moreover, several other cleaning processes have been employed, as described in the Athens' case study above. After cleaning, aggregation and filtering the resulting data show acceptable reliability scores (overall above 0.4 in a scale from 0 to 1, where 1 is the highest reliability), and then it was necessary to transform the loop detector data and split in three distinct datasets, for speed, flow and occupancy respectively. Moreover, resampling of the dataset every 5 minutes has been performed.

##### *9.1.2.1.1 Locations of the sensors*

For the city of Antwerp, there is a multitude of loop detectors available which provide one-minute readings regarding the network conditions, as shown in Figure 9-10, - a total of 4478 to be precise. However, the area which has been deemed suitable to be implemented and tested for the use case of incident detection represents the corridor between the port of Antwerp and Eindhout including E313 motorway in both directions. This area includes 103 nodes in Direction 1 and 164 nodes in Direction 2 and in Figure 9-9 the locations of the sensors of the model are depicted on a map.

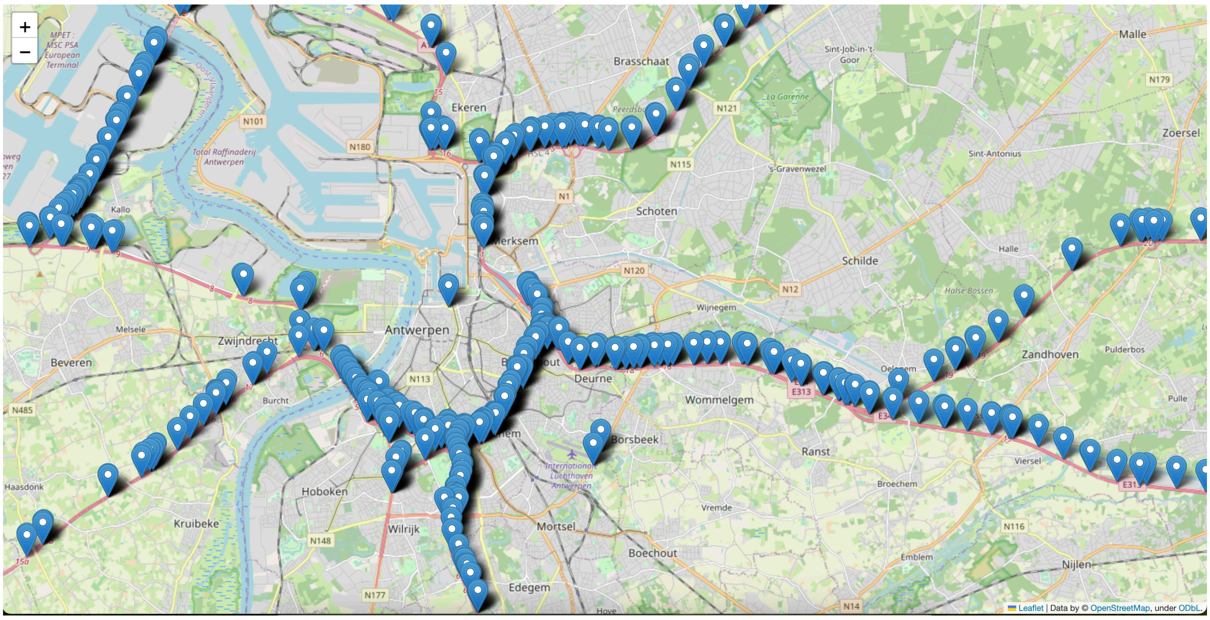


**Figure 9-10: Available loop detectors around the Antwerp area.**

### *9.1.2.1.2 Data collection*

Traffic data measured from loop detectors from highways in the region of Flanders are updated each minute, this data is available online[9] and is presented in XML format. The data is collected from our part every minute, the data labels are adapted accordingly to Datex II and finally we publish the data in Orion and store it in the Data Storage.

### *9.1.2.1.3 Raw data characteristics*

The traffic data for E313 highway collected by loop detectors include the number of vehicles and the average speed, occupancy and flow, in addition to other traffic measurements for 5 different vehicle classes/categories, aggregated per minute and the location of the measurement points.

Traffic data updated each minute from highways in the region of Flanders is available at http://miv.opendata.belfla.be/miv/verkeersdata.The description of the

[9] thttp://miv.opendata.belfla.be/miv/verkeersdata

data, including the original tag and the data description from the data source is summarized in Table 9-2.

**Table 9-2: Data information for the MIV loop-based traffic data elements.**

| Tag | Description |
|---|---|
| meetpunt | Data per measurement point under the element "meetpunt".<br>• unieke_id: Unique identification number of the measurement point. More data (location, etc.) about the measurement point are found in the configuration: http://miv.opendata.belfla.be/miv/configuratie/xml<br>beschrijvende_id: Descriptive id. (Internally used id. May be omitted in the future.) |
| lve_nr | Number of the LVE (Local Processing Unit). The LVE processes the data of a group of measurement points This number is used internally. This data can be omitted in the future.) |
| tijd_waarneming | "Obervation time". Starting date and time of the minute to which the data correspond, UTC+1. the date is several years in the past, this can point to a restarted measurement device which hasn't synchronised its time yet. that case, if the data still changes every minute, it can be assumed that the data is live and current. pe. 13:00:00 contains the minute between 13:00:00 and 13:00:59 |
| tijd_laatst_gewijzigd | Date and time of the last update of data for this measurement point. |
| actueel_publicatie | • 0 = Data of this point has a tijd_waarneming older than about 3 minutes ago<br>1 = Data of this point has a tijd_waarneming more recent than about 3 minutes ago. This might signify connection problems. The measurement point may be offline. |
| beschikbaar | Availability of the measurement point:<br>• 0 = The measurement point is currently unavailable<br>• 1 = The measurement point is currently available |
| defect | Failure-status of the measurement point<br>• 0 = no failure |

| | |
|---|---|
| | • 1 = 1 of both detection loops is probably failing<br><br>2 = more than 20 percent bad counts: severe failure |
| geldig | Regularity of the data:<br><br>• 0 = Regular data<br><br>• 1 = Indication for irregular data<br><br>• 2 = Indication for extremely irregular data<br><br>3 = Indication for extremely irregular data caused by a failure |
| meetdata | Data measurement for each type of vehicle class |
| verkeersintensiteit | Vehicle count within vehicle class. |
| voertuigsnelheid_rekenkundig | Sum (vi) / n = arithmetic average speed of the vehicles in this vehicle class (with vi = individual speed of a vehicle in this vehicle class)<br><br>• Value domaing 0 to 254 km/h.<br><br>• Value range 0..200 km/h<br><br>• Resolution 1.<br><br>Special values:<br><br>• 251: Initial value<br><br>• 254: Calculation not possible<br><br>252: no vehicles were counted in this vehicle class. |
| voertuigsnelheid_harmonisch | n / Sum (1/vi) = harmonic average speed of the vehicles in this vehicle class with vi = individual speed of a vehicle in this vehicle class)<br><br>Special values:<br><br>• 251: Initial value<br><br>• 254: Calculation not possible<br><br>252: no vehicles were counted in this vehicle class. |
| klasse_id | Vehicle class. More information about each class can be found in Table 9-3. |
| rekendata | • Calculated data |

| | |
|---|---|
| bezettingsgraad | Occupancy = Pointcoverage (in 10ms) / 60s (in sec)<br><br>• Pointcoverage: Time during which a fictional point of the detector was covered by a vehicle. The pointcoverage is expressed in units of 10 milliseconds |
| beschikbaarheidsgraad | Degree of availability = ((60s - unavailability) / 60s) * 100 Unavailability = Time during which a detector was unable to reliably detect passing vehicles. |
| onrustigheid | Sum ($vi^2$) / N - ( sum (vi) / N $)^2$ (including all vehicles from all classes)<br><br>• (vi) = speed of vehicle i<br><br>N = total vehicle count |

**Table 9-3: Vehicle classes/categories for Antwerp traffic dataset.**

| **Vehicle Class number** | **Description** |
|---|---|
| Vehicle class 1 | This vehicle class was used for vehicles with estimated length between 0m and 1,00m. Pe. motorbikes. The occasional measurements in this vehicle class are unreliable. This data is unused by AWV and the Traffic Center. |
| Vehicle class 2 | Cars = vehicles with an estimated length between 1,00m and 4,90m |
| Vehicle class 3 | Vans = vehicles with an estimated length between 4,90m and 6,90m |
| Vehicle class 4 | Rigid lorries = vehicles with an estimated length between 6,90m and 12,00mbv.: Lorry, or tractor |
| Vehicle class 5 | (Semi-)Trailers or busses= vehicles with an estimated length longer than 12,00m by.: lorry with trailer, tractor with semi-trailer, or bus. |

To ensure the minimum changes over time, we have avoided to use the tags labelled as “internal use only” or “Can be omitted in the future”, such as lve_nr and beschrijvende_id. Furthermore, the original tags have been translated to DATEX II.

**Table 9-4: Mapping between XML and DATEX II tags. DATEX II tag's descriptions are also included.**

| XML Tag | DATEX II tag |
|---|---|
| tijd_waarneming | timeValue |
| tijd_laatst_gewijzigd | lastUpdateOfDeviceInformation |
| actueel_publicatie | lastDeviceCheck |
| beschikbaar | statusDescription |
| defect | faultDescription |
| geldig | regularity |
| meetdata | MeasuredData |
| verkeersintensiteit | vehicleFlowRate |
| voertuigsnelheid_rekenkundig | averageVehicleSpeed |
| voertuigsnelheid_harmonisch | harmonicSpeed |
| klasse_id | stationType / vehicleType |
| rekendata | ElaboratedData |
| bezettingsgraad | occupancy |
| beschikbaarheidsgraad | availabilityRate |
| onrustigheid | restlessness |

For real time updates about incidents, traffic flow, roads status and events affecting traffic on the highways in Flanders, we have collected and used data from https://www.verkeerscentrum.be/uitwisseling/datex2v3, presented in xml. This data is already published using DATEX II tags and the description of the data is described in Table 9-5. In this case, thus, there is no need to modify any tag.

**Table 9-5: Data information for the MIV real time traffic information.**

| DATEX-II Tag | Description |
|---|---|
| situation | An identifiable instance of a traffic/travel situation comprising one or more traffic/travel circumstances which are linked by one or more causal relationships. Each traffic/travel circumstance is represented by a Situation Record.<br><br>Id: Situation Id |
| situationVersionTime | The status of the related information (real, test, exercise ....). |
| headerInformation | Management information relating to the data contained within a publication. |
| confidentiality | The extent to which the related information may be circulated, according to the recipient type. |
| informationStatus | The status of the related information (real, test, exercise ....). |
| situationRecord | An identifiable versioned instance of a single record/element within a situation.<br><br>• Type: Type of the situation record.<br><br>• Id: Id of the situation record.<br><br>Version: Version of the situation record |
| situationRecordCreationTime | The date/time that the SituationRecord object (the first version of the record) was created by the original supplier. |
| situationRecordVersionTime | The date/time that this current version of the SituationRecord within the situation was written into the database of the supplier which is involved in the data exchange. Identity and version of record are defined by the class stereotype implementation. |
| probabilityOfOccurrence | An assessment of the degree of likelihood that the reported event will occur. |
| safetyRelatedMessage | Indicates, whether this SituationRecord specifies a safety related message according to Commission Delegated Regulation (EU) No 886/2013. |
| validity | Specification of validity, either explicitly or by a validity time period specification which may be discontinuous. |
| validityStatus | Specification of validity, either explicitly overriding the validity time specification or confirming it. |
| validityTimeSpecification | A specification of periods of validity defined by overall bounding start and end times and the possible intersection of valid periods with exception periods (exception periods overriding valid periods). |

| | |
|---|---|
| overallStartTime | Start of bounding period of validity defined by date and time. |
| overallEndTime | End of bounding period of validity defined by date and time. |
| locationReference | The location (e.g. the stretch of road or area) to which the data value(s) is or are pertinent/relevant. This may be different from the location of the measurement equipment (i.e. the measurement site location).<br><br>Type: Type of the location reference |
| complianceOption | Defines whether the network management instruction or the control resulting from a network management action is advisory or mandatory. |
| roadOrCarriagewayOrLaneManagementType | Type of road, carriageway or lane management action instigated by operator. |
| pointByCoordinates | A single point defined only by a coordinate set with an optional bearing direction. |
| pointCoordinates | A pair of planar coordinates defining the geodetic position of a single point using the European Terrestrial Reference System 1989 (ETRS89). |
| latitude | Latitude in decimal degrees using the European Terrestrial Reference System 1989 (ETRS89). |
| longitude | Longitude in decimal degrees using the European Terrestrial Reference System 1989 (ETRS89). |
| alertCPoint | A collection of information describing locations using the Alert-C location referencing approach.<br><br>Type: Type of alertCPoint |
| alertCLocationCountryCode | Country code from the alert location |
| alertCLocationTableNumber | Number allocated to an ALERT-C table in a country. Ref. EN ISO 14819-3 for the allocation of a location table number. |
| alertCLocationTableVersion | Version number associated with an ALERT-C table reference. |
| alertCDirection | The direction of traffic flow along the road to which the information relates. |
| alertCDirectionCoded | Direction of navigation with respect to secondary to primary location (RDS direction) |
| alertCMethod4PrimaryPointLocation | The point (called Primary point) which is either a single point or at the downstream end of a linear road section. The point is specified |

| | |
|---|---|
| | by a reference to a point in a pre-defined ALERT-C location table plus a non-negative offset distance. |
| alertCMethod4SecondaryPointLocation | The point (called Primary point) which is either a single point or at the downstream end of a linear road section. The point is specified by a reference to a point in a pre-defined ALERT-C location table plus a non-negative offset distance. |
| alertCLocation | Identification of a specific point, linear or area location in an ALERT-C location table. |
| specificLocation | Unique code within the ALERT-C location table which identifies the specific point, linear or area location. |
| offsetDistance | The non-negative offset distance from the ALERT-C referenced point to the actual point. The ALERT-C locations in the primary and secondary locations must always encompass the linear section being specified, thus offset distance is towards the other point. |
| gmlLineString | Line string based on GML (EN ISO 19136) definition: a curve defined by a series of two or more coordinate tuples. Unlike GML may be self-intersecting.<br><br>SrsName: Source name if this is not present, posList is assumed to use "ETRS89-LatLonh" reference system |
| posList | List of coordinate Tuples define the geometry of this GmlLineString. There must be at least 2 Tuples of coordinates. |
| alertCLinear | A linear section along a road defined between two points on the road by reference to a pre-defined ALERT-C location table.<br><br>Type: Type of alertCLinear |

```
                                              id  entityVersion  \
0  url:schema:Frontier:Flanders:MeasurementPoint:29              2
1  url:schema:Frontier:Flanders:MeasurementPoint:30              2
2  url:schema:Frontier:Flanders:MeasurementPoint:31              2
3  url:schema:Frontier:Flanders:MeasurementPoint:32              2
4  url:schema:Frontier:Flanders:MeasurementPoint:33              2

   lastDeviceCheck                  timeValue lastUpdateOfDeviceInformation  \
0                1  2022-10-26T13:35:00+01:00     2022-10-26T14:36:14+02:00
1                1  2022-10-26T13:35:00+01:00     2022-10-26T14:36:14+02:00
2                1  2022-10-26T13:35:00+01:00     2022-10-26T14:36:14+02:00
3                1  2022-10-26T13:35:00+01:00     2022-10-26T14:36:14+02:00
4                1  2022-10-26T13:35:00+01:00     2022-10-26T14:36:14+02:00

   statusDescription  faultDescription  regularity  \
0                  0                 0           0
1                  0                 0           0
2                  0                 0           0
3                  0                 0           0
4                  0                 0           0
```

**Figure 9-11: Exploration of the raw traffic data for Antwerp.**

For the context of our case, we have transformed the data using DATEX II attribute tags and the dataframe created from reading the transformed XML is illustrated in Figure 9-11. As part of the dataset for each sample, here are also two attributes which contain nested data: the *measured* and *elaborated* constructs. An instantiation of *measured* attribute as part of a sample is shown below:

[{“vehicleClass”: 1, “vehicleFlowRate”: 0, “averageVehicleSpeed”: 0, “harmonicSpeed”: 252}, {“vehicleClass”: 2, “vehicleFlowRate”: 0, “averageVehicleSpeed”: 0, “harmonicSpeed”: 252}, {“vehicleClass”: 3, “vehicleFlowRate”: 0, “averageVehicleSpeed”: 0, “harmonicSpeed”: 252}, {“vehicleClass”: 4, “vehicleFlowRate”: 0, “averageVehicleSpeed”: 0, “harmonicSpeed”: 252}, {“vehicleClass”: 5, “vehicleFlowRate”: 0, “averageVehicleSpeed”: 0, “harmonicSpeed”: 252}]

whereas for the elaborated, an example is shown below:

{"occupancy": 0, "availabilityRate": 100, "restlessness": 0}

#### *9.1.2.1.4 Data cleaning and filtering*

In the pre-processing phase of our analysis, we employed a meticulous filtering and cleaning process to ensure the integrity and applicability of the loop detector data. Initially, we identified and rectified any anomalies in the data, such as outliers or incomplete records. Subsequently, to streamline the dataset for more coherent analysis, we aggregated the different vehicle categories into a single, consolidated attribute. This means that we only kept the sum of vehicleFlowRate and the average of the speed, in addition to the supplementary attributes mentioned above. This aggregation enables a more generalized assessment of traffic patterns while maintaining the robustness of the data. Moreover, the dataset has been harmonized to correspond with the corresponding data model.

#### *9.1.2.1.5 Cleaned data characteristics*

The cleaning and filtering process, as described in the previous section, has resulted in a dataframe which contains the id of each sensor included the dataset, the timeValue of the measurements, the averageVehicleSpeed (representing the speed of the measured timestamp in that particular loop detector), the vehicleFlowRate and occupancy(representing the flow of vehicles and occupancy respectively, measured by the loop detector in that specific timestamp). Moreover, the attributes of availabilityRate, regularity, faultDescription and statusDescription are included in the dataframe. Figure 9-12 shows the final dataset after the cleaning, imputation and aggregation process has been done.

| | id | timeValue | averageVehicleSpeed | vehicleFlowRate | occupancy | availabilityRate | regularity | faultDescription | statusDescription |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 139 | 2022-10-26T13:35:00+01:00 | 0.0 | 0.0 | 0 | 100 | 0 | 0 | 1 |
| **1** | 140 | 2022-10-26T13:35:00+01:00 | 35.2 | 1080.0 | 14 | 100 | 0 | 0 | 1 |
| **2** | 141 | 2022-10-26T13:35:00+01:00 | 50.8 | 1260.0 | 14 | 100 | 0 | 0 | 1 |
| **3** | 142 | 2022-10-26T13:35:00+01:00 | 25.2 | 1680.0 | 11 | 100 | 0 | 0 | 1 |
| **4** | 143 | 2022-10-26T13:35:00+01:00 | 20.2 | 360.0 | 10 | 100 | 0 | 0 | 1 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **2080918** | 4203 | 2022-10-31T23:57:00+01:00 | 37.4 | 360.0 | 2 | 100 | 0 | 0 | 1 |
| **2080919** | 4204 | 2022-10-31T23:57:00+01:00 | 0.0 | 0.0 | 0 | 100 | 0 | 0 | 1 |

**Figure 9-12: Cleaned data characteristics for Antwerp dataset.**

#### *9.1.2.1.6 Data transformation*

After the cleaning and filtering process, it was necessary to transform the loop detector data and split the dataset in three distinct datasets, for speed, flow and occupancy respectively. Moreover, resampling of the dataset every 5 minutes has been performed. The datasets are shown in Figure 9-13, Figure 9-14 and Figure 9-15.

```
id                           1104    1105    1106    1107      1112    1113  \
timeValue
2022-10-27 00:00:00+01:00  19.200000   0.00  42.88   0.00  16.280000   4.68
2022-10-27 00:05:00+01:00  33.560000   5.16  22.56   0.00  34.320000   9.96
2022-10-27 00:10:00+01:00  31.040000   9.24  23.08   4.80  41.240000   4.24
2022-10-27 00:15:00+01:00  15.120000   4.72  34.20   9.88  19.840000   3.84
2022-10-27 00:20:00+01:00  19.320000   5.16  10.32   0.00  22.760000   0.00
...                              ...    ...    ...    ...        ...    ...
```

**Figure 9-13: Transformation of traffic dataset of Antwerp - speed.**

```
id                          1104   1105   1106   1107   1112   1113   1114  \
timeValue
2022-10-27 00:00:00+01:00   75.0    0.0  192.0    0.0   60.0   12.0   75.0
2022-10-27 00:05:00+01:00  156.0   12.0   72.0    0.0  144.0   24.0  132.0
2022-10-27 00:10:00+01:00  120.0   36.0  108.0   12.0  156.0   12.0  180.0
2022-10-27 00:15:00+01:00   48.0   12.0  144.0   24.0   60.0   12.0   36.0
2022-10-27 00:20:00+01:00   84.0   12.0   24.0    0.0   84.0    0.0   84.0
...                          ...    ...    ...    ...    ...    ...    ...
```

**Figure 9-14: Transformation of traffic dataset of Antwerp - flow.**

```
id                          1104  1105      1106  1107  1112      1113  \
timeValue
2022-10-27 00:00:00+01:00    1.0   0.0  1.600000   0.0   0.8  0.000000
2022-10-27 00:05:00+01:00    1.0   0.0  0.400000   0.0   1.0  0.000000
2022-10-27 00:10:00+01:00    1.2   0.0  0.400000   0.0   1.2  0.000000
2022-10-27 00:15:00+01:00    0.6   0.0  1.000000   0.0   0.6  0.000000
2022-10-27 00:20:00+01:00    0.8   0.0  0.200000   0.0   1.0  0.000000
...                          ...   ...       ...   ...   ...       ...
```

**Figure 9-15: Transformation of traffic dataset of Antwerp - occupancy.**

Moreover, in order to align the loop detector dataset with the structure of the respective incidents dataset, which involves segments and not point locations, we utilized the Geopandas library to spatially join the point-based detector data with the segment-based incident records. This geospatial analysis required precise mapping of loop detector coordinates, as provided by stakeholder operators in corresponding CSV files, to the predefined road segments where incidents were catalogued. The alignment process ensured that each loop detector's data was accurately associated with the corresponding road segment, facilitating a direct comparison between traffic conditions and incident occurrences. For visualization purposes, and to validate the accuracy of our transformation, we employed the Folium library to map the loop detectors onto an interactive map, overlaying this with the incident segments to confirm the correctness of the alignment. This meticulous approach enabled a robust spatial analysis, ensuring that traffic patterns could be analyzed within the exact context of incident locations. The point-based dataset has thus been transformed to 82 respective segments. (41 per direction), as illustrated in Figure 9-16.

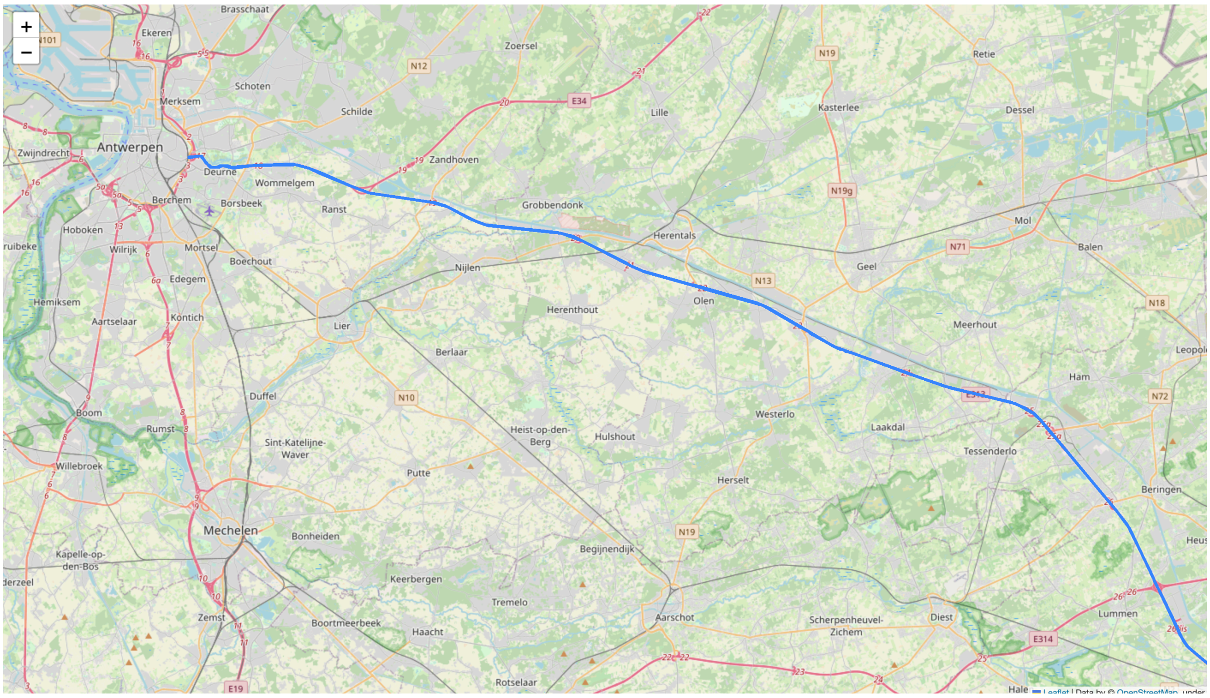



**Figure 9-16: Mapping of loop detectors to segments for Antwerp.**

The final format of the three created distinct datasets for speed, flow and occupancy is illustrated in respectively.

| timeValue | 88004568 | 2347002507 | 2453002822 | 2507002627 | \ |
|---|---|---|---|---|---|
| 2022-10-27 00:00:00+01:00 | 17.89 | 8.24 | 28.38 | 9.33 | |
| 2022-10-27 00:05:00+01:00 | 21.63 | 13.38 | 21.36 | 10.27 | |
| 2022-10-27 00:10:00+01:00 | 18.02 | 12.14 | 11.28 | 6.95 | |
| 2022-10-27 00:15:00+01:00 | 20.10 | 17.22 | 13.50 | 12.32 | |
| 2022-10-27 00:20:00+01:00 | 15.53 | 16.66 | 12.72 | 12.11 | |
| ... | ... | ... | ... | ... | |

**Figure 9-17: Final traffic dataset for Antwerp using segments - speed.**

| timeValue | 88004568 | 2347002507 | 2453002822 | 2507002627 | \ |
|---|---|---|---|---|---|
| 2022-10-27 00:00:00+01:00 | 122.67 | 24.0 | 102.0 | 33.00 | |
| 2022-10-27 00:05:00+01:00 | 150.67 | 48.0 | 114.0 | 36.00 | |
| 2022-10-27 00:10:00+01:00 | 112.00 | 36.0 | 37.5 | 21.00 | |
| 2022-10-27 00:15:00+01:00 | 129.33 | 72.0 | 66.0 | 54.00 | |
| 2022-10-27 00:20:00+01:00 | 113.33 | 60.0 | 36.0 | 45.75 | |
| ... | ... | ... | ... | ... | |

**Figure 9-18 Final traffic dataset for Antwerp using segments - flow.**

```
                            88004568   2347002507   2453002822   2507002627  \
timeValue
2022-10-27 00:00:00+01:00       0.80         0.10         0.40         0.10
2022-10-27 00:05:00+01:00       1.18         0.60         0.80         0.45
2022-10-27 00:10:00+01:00       0.87         0.40         0.12         0.25
2022-10-27 00:15:00+01:00       0.89         0.70         0.40         0.35
2022-10-27 00:20:00+01:00       0.69         0.40         0.20         0.35
...                              ...          ...          ...          ...
```

**Figure 9-19 Final traffic dataset for Antwerp using segments - occupancy.**

### *9.1.3 Labelled Incidents Dataset*

The labelled incidents dataset is extremely important for the incident detection task's evaluation, since it constitutes the ground truth on which the performance metrics of our models will be based.

#### 9.1.3.1 Athens Case Study

##### *9.1.3.1.1 Data collection*

The end-user partner, Attikes Diadromes, has provided us with an Excel file containing information about the incidents which had been registered in Attiki Odos from October 2020 until April 2022. This dataset is used in conjunction with the corresponding historical data from IDL sensors obtained from Attiki Odos, in order to build the initial data-driven models for incident detection in this use case.

Regarding real-time collection of incidents, the users/operators are able to insert incidents happening in real-time in the system through the dashboard developed, thus this constitutes another way of integrating new incidents and fusing those within this dataset.

##### *9.1.3.1.2 Raw data characteristics*

As explained in Chapter 9.1.1.1.16, the historical labelled incidents dataset provided for the task of incident detection consists in an excel file with 34652 observations and 34 feature columns. Columns were translated to English from Greek to grasp the meaning of each tagged feature. The file was converted to a Python pandas data frame for exploratory analysis, and a unique values' analysis was carried out. The main source of labelled incidents came from a closed-circuit television (CCTV) set (a total of 6489), branches A and E had the highest amount of recorded incidents (13829 and 13757 respectively), pk-points were annotated in the dataset (but there is a lack of loop detectors specifications), start and end time of the incident was annotated together with the queue length in number of cars. However, some inconsistencies were also noted, for instance, incidents that had no impact to traffic also appeared as labelled incident.

In the following figures, some of the outcomes of the Exploratory Data Analysis regarding the timespan of the data can be found:

| year | month | count |
|---|---|---|
| 2020 | 10 | 1839 |
| | 11 | 1138 |
| | 12 | 1148 |
| 2021 | 1 | 1372 |
| | 2 | 1244 |
| | 3 | 1314 |
| | 4 | 1524 |
| | 5 | 2065 |
| | 6 | 2534 |
| | 7 | 2687 |
| | 8 | 2100 |
| | 9 | 2141 |
| | 10 | 2106 |
| | 11 | 2107 |
| | 12 | 2132 |
| 2022 | 1 | 1598 |
| | 2 | 1814 |
| | 3 | 1767 |
| | 4 | 2020 |

**Figure 9-20: Traffic incident distribution per year and per month.**

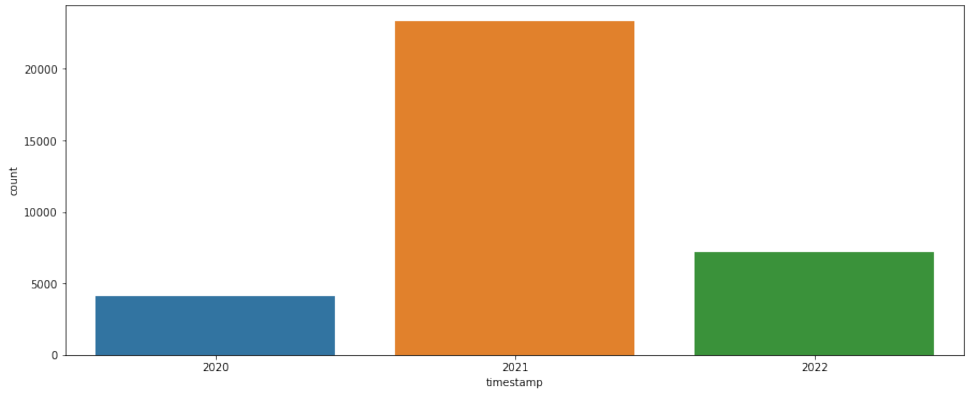



**Figure 9-21: Yearly distribution of traffic incidents.**

Below in Figure 9-22, the histogram is showing the distribution of incidents in May 2021, where each bar represents the number of incidents on a particular day.

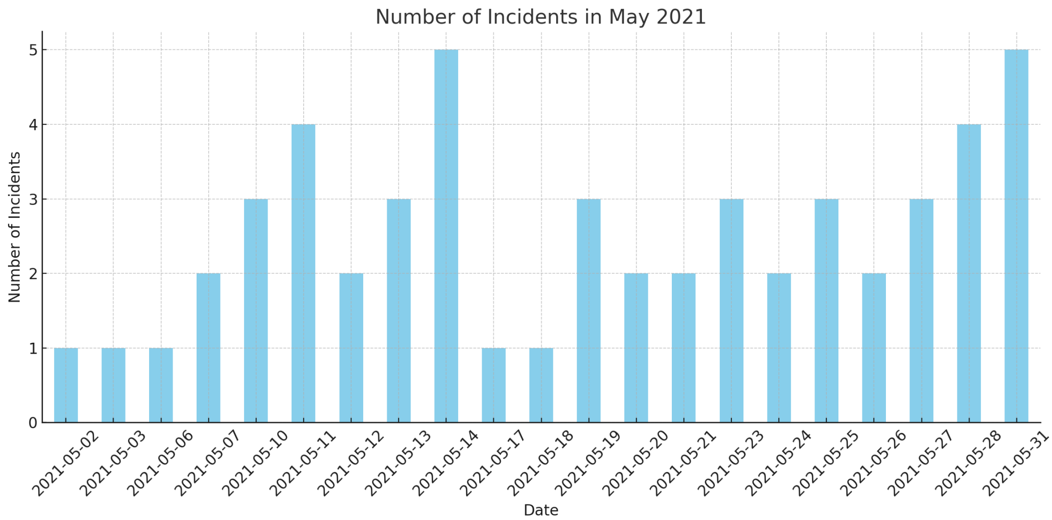



**Figure 9-22: Distribution of number of reported incidents in Attiki Odos - May 2021.**

#### *9.1.3.1.3 Data cleaning and filtering*

As soon as we have managed to collect the data and derived some first impressions on them through Exploratory Data Analysis, the next step was to perform data cleaning.

One of the columns of this dataset referred to the timestamp of the recorded incident. We have observed that two observations between January and October 2020 were included, which was probably due to an error in the dataset collection, therefore, these two observations were ultimately discarded.

Then, in order to proceed with the filtering of the incidents which actually had an impact on the traffic state of the network, we examined correlations between different features/columns. We have tried various combinations, though in the next paragraphs we are going to briefly explain the most significant ones, which were included in the filtering process.

From Figure 9-23, we have observed that he vast majority of the observed incidents stemmed from the CCTV cameras. (For completeness reasons, we state that the sources as illustrated on the Y-axis also include 1024 telephone number, Emergency Roadside Telephones (ERT), traffic police, Interamerican and others).

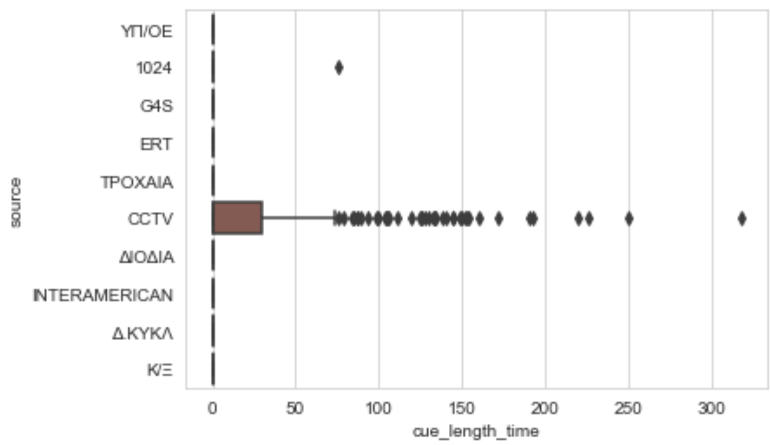



**Figure 9-23: Queue length time in relation to the information source of incidents.**

From Figure 9-24 and Figure 9-25, we observe that most of the observed incidents belong to the type ***1 - Traffic Congestion*** and ***4D - Traffic accident*** ,and that the columns containing information about *the queue length time* and *queue length of the cars* demonstrate a correlation between them. All types of incidents present in the labelled incidents dataset of the Athens use case are shown in Table 9-6.

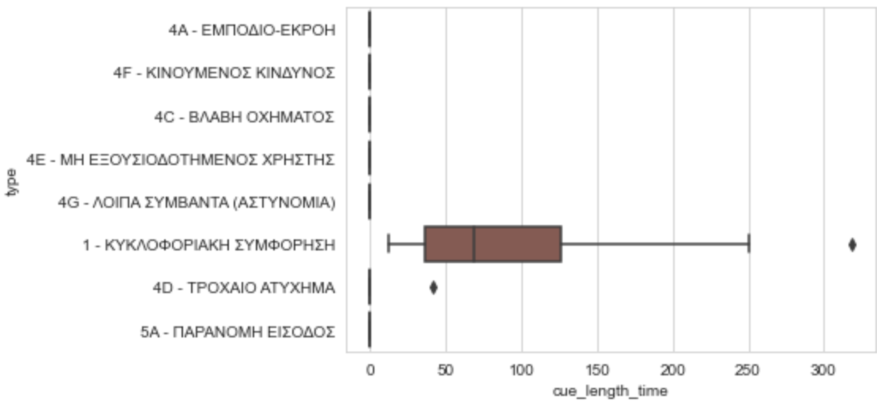



**Figure 9-24: Queue length time in relation to the incident type.**

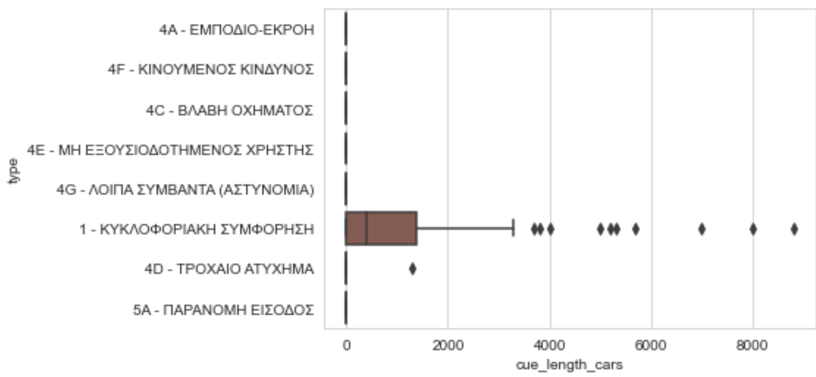



**Figure 9-25: Queue length of cars (in meters) in relation to the incident type.**

**Table 9-6: Types of incidents in the Athens labelled incident dataset.**

| TYPE | SUBTYPE |
|---|---|
| 1 - TRAFFIC CONGESTION | 1.A - DUE TO AN INCIDENT |
| | 1.B - DUE TO TRAFFIC LOAD |
| 2 - EXTREME WEATHER EVENTS | 2.A - HEAVY RAIN |
| | 2.B - SNOW |
| 4A - OBSTRUCTION-OUTFLOW | 4A.A - OUTFLOW |
| | 4A.B - DEAD OR INJURED ANIMAL |
| | 4A.C - LARGE OBJECT |
| 4B - ABANDONED VEHICLE | 4B. ABANDONED VEHICLE |
| 4C - VEHICLE FAILURE | 4C.A - MECHANICAL FAILURE |
| | 4C.B - FUEL |

| | |
|---|---|
| | 4C.C - TIRES |
| 4D - TRAFFIC ACCIDENT | 4D.X - LEFT DEFLECTION |
| | 4D.D - NTOMETOPIC COLLISION |
| | 4D.Ω1 - IMPACT ON TOLL EQUIPMENT |
| | 4D.O - OVERTURN ON THE ROAD |
| | 4D.B - SIDE-FRONTAL COLLISION |
| | 4D.Ω2 - OBSTACLE IMPACT |
| | 4D.C - SIDE COLLISION |
| | 4D.N - RIGHT COLLISION |
| | 4D.Ω3 - DROPPING OF ITEMS FROM THE FRONT |
| | 4D.P - FIRE |
| | 4D.T - COLLISION OF 3 VEHICLES (KARAMBOLA) |
| | 4D.X - IMPACT ON PERMANENT MARKING |
| | 4D.Y - IMPACT ON A TEMPORARY MARKING |
| | 4D.L - DRIFTING OF ANIMAL |
| | 4D.F - ON A PARKED VEHICLE |
| | 4D.Y - COLLISION OF 4 VEHICLES (KARAMBOLA) |
| | 4D.Ω4 - OUTFLOW FROM A VEHICLE IN FRONT |
| | 4D.Z - ON A VEHICLE THAT MAKES A FORCED STOP |
| | 4D.S -OTHER |
| | 4D.TH - IN PILLAR OR TREE |
| | 4D.Y1 - COLLISION OF MORE THAN FOUR VEHICLES (KARABOLA) |
| 4E - UNAUTHORIZED USER | 4E.A - PEDESTRIAN |
| | 4E.C - EXCESS HEIGHT |
| | 4E.D - LOW SPEED |
| | 4E.E - OTHER |
| | 4E.B - CYCLIST |

| | |
|---|---|
| | 4E.F - OTHER |
| 4F - DANGEROUS RISK | 4F.C - MOVING ANIMAL |
| | 4F.B - POOR LOADING |
| | 4F.A – DIFFERENT TRAFFIC DIRECTION |
| | 4F.D - OTHER |
| 4G - OTHER EVENTS (POLICE) | 4G.A - BOMBING THREAT |
| | 4G.C - OTHER CAUSE |
| 4L - FIRE AT KEP | 4L.A - TECHNICAL DEFECT |
| 5A - ILLEGAL ENTRY | 5A.A - AT A TOLL STATION |

We have also examined the distribution of the queue length of cars across different ranges of values. We can see that the vast majority are concentrated between 0 and 1500, but there are several outliers which expand until 12000 meters, as shown in the Figure 9-26 below:

```
(-0.001, 100.0]       614
(100.0, 200.0]        113
(200.0, 300.0]         73
(300.0, 400.0]         67
(400.0, 500.0]         82
(500.0, 1000.0]       228
(1000.0, 1500.0]       63
(1500.0, 2000.0]       82
(2000.0, 3000.0]      132
(3000.0, 4000.0]      100
(4000.0, 5000.0]       63
(5000.0, 10000.0]     126
(10000.0, 12000.0]     14
Name: cue_length_cars, dtype: int64
```

**Figure 9-26: Distribution of queue length of cars in intervals.**

Based on the above constatations, we have decided to filter the incidents for the initial preliminary experiments according to the following rules:

- We have filtered the incidents based on their type, namely ***1 - Traffic Congestion*** and ***4D - Traffic accident*** and the rest of the values were not included in the labelled incidents dataset due to the fact that they do not affect the road and therefore will not be captured in our preliminary experiments.

- Finally, the incidents have been filtered on the queue length of the cars observed.
    - As shown in the above Exploratory Data Analysis regarding the percentile distribution of the queue length, we identified 200 meters as a threshold for the filtering.
    - After communication with stakeholders, we received feedback to reduce the threshold for queue length size to 50 (instead of 200 m. which was our initial proposal). This proposal was based on their expertise and justified by the fact that it is extremely rare to have queues of 200 meters in Attiki Odos, even when an unplanned incident would occur.

##### *9.1.3.1.4 Data transformation*

After the data cleaning and the filtering, the dataset needed to be transformed in a way that it could be fed in the Machine Learning algorithms, as the labels in case of Supervised training, or for evaluation in the case of Unsupervised methods.

Based on the start and end location of the incidents which was specified in the relative columns of the dataset, we have created a transformer script which maps these locations to the closest sensor Unit IDs. This permits us to have a mapping between the incident dataset and the flow, speed and occupancy observations gathered by the loop sensors, based on the id of each of those. There were difficulties involved in the sense that for the case of junctions, it has been quite complex to identify the IDs of the sensors in the affected areas, since there was no direct link between the exact location and several calculations were required to get an approximation. Finally, we were able to transform the filtered incidents dataset for branches A and E for the month of May in 5-minute intervals where rows refer to timestamps and columns were the ids of the loop sensors, and the values of the matrix were 1 if this location and time corresponds to an incident occurrence, or 0 otherwise.

##### *9.1.3.1.5 Limitations*

We would like to specify the fact that there are limitations to the presented steps, specifically regarding data filtering, along with inherent limitations stemming from the collected data itself, some of which are briefly discussed below:

- It is possible that some of the accidents are not registered. From the analysis, all of the incidents have been recorded manually and most of them have been identified through CCTV cameras.
- Another possibility is that perhaps the recorded timings are not accurate. It is possible that an event which had happened on a timestamp *t*, is recorded on a later timestamp. However, this has a severe impact on the evaluation of our algorithms.
- Last but not least, the process which we have employed for filtering could be prone to errors. The selection criteria were based on stakeholders' expertise and the dataset characteristics, albeit there is a possibility that some incidents which had a severe impact to the traffic state are neglected and not taken into account.

#### 9.1.3.2 Antwerp Use Case

Together with the loop detector dataset which contain information of the traffic characteristics, we have managed to acquire an Excel file with details for the incidents which had been registered in E313 Antwerp highway. The labelled incidents dataset includes 5774 incident occurrences in total, spanning from 2011-04-26 11:29:19.647 until 2023-08-29 13:29:52.907 and more specifically for our case and the dates selected (2022-10-27 to 2023-08-29), it contains 526 incidents for 2023 and 136 incidents for 2022. This dataset contains the following fields: *segment_id, incident_id, registration_time, duration, direction* and location of the incident (*kmpt1* and *kmpt2*). Based on the start timestamp and duration of the incidents specified in the relative columns of the dataset, we have mapped these timestamps to the corresponding start and end time. However, it is impossible to perform further filtering of this dataset based on its impact in traffic or severity from the information available. Thus, all the incidents reported in the dataset are kept in the evaluation dataset. Finally, the filtered incidents dataset has been transformed in 5-minute intervals where rows refer to timestamps and columns were the ids of the segments and the values of the matrix were 1 if this location and time corresponds to an incident occurrence, or 0 otherwise.

#### *9.1.3.2.1 Data collection*

The respective data provider from Verkeercentrum provided us with an Excel file containing information about the incidents which had been registered in E313 Antwerp highway. This dataset is used in conjunction with the corresponding historical data from ILD sensors obtained.

#### *9.1.3.2.2 Data characteristics*

The labelled incidents dataset includes 5774 incident occurrences in total, spanning from 2011-04-26 11:29:19.647 until 2023-08-29 13:29:52.907 and more specifically for our case, it contains 110 incidents for 2023 and 530 incidents for 2022.

| sg_id | incident_number | registration_time | duration | segment_id | ident8 | kmpt1 | kmpt2 | shape | sl |
|---|---|---|---|---|---|---|---|---|---|
| 8004568 | 6158490 | 2019-05-21 15:39:24.907 | 89.17 | 88004568 | A0130001 | 0.0 | 0.491 | 0x8A7A000001041700000000083C04A2AFA024100234A7B... | LIN (155461.286 211752.68 |
| 8004568 | 7869905 | 2022-02-23 12:28:34.740 | 75.05 | 88004568 | A0130001 | 0.0 | 0.491 | 0x8A7A000001041700000000083C04A2AFA024100234A7B... | LIN (155461.286 211752.68 |
| 8004568 | 2148535 | 2015-11-23 05:27:52.947 | 88.58 | 88004568 | A0130001 | 0.0 | 0.491 | 0x8A7A000001041700000000083C04A2AFA024100234A7B... | LIN (155461.286 211752.68 |
| 8004568 | 4577054 | 2017-12-17 10:35:32.080 | 25.22 | 88004568 | A0130001 | 0.0 | 0.491 | 0x8A7A000001041700000000083C04A2AFA024100234A7B... | LIN (155461.286 211752.68 |
| 8004568 | 4671928 | 2018-01-28 06:54:17.827 | 34.89 | 88004568 | A0130001 | 0.0 | 0.491 | 0x8A7A000001041700000000083C04A2AFA024100234A7B... | LIN (155461.286 211752.68 |

**Figure 9-27: Sample of incident dataset for Antwerp.**

```
           sg_id  incident_number       registration_time  duration ident8  \
0       88004568          6158490 2019-05-21 15:39:24.907     89.17      1
1       88004568          7869905 2022-02-23 12:28:34.740     75.05      1
2       88004568          2148535 2015-11-23 05:27:52.947     88.58      1
3       88004568          4577054 2017-12-17 10:35:32.080     25.22      1
4       88004568          4671928 2018-01-28 06:54:17.827     34.89      1
...          ...              ...                     ...       ...    ...
5877  3848003084          9259959 2023-07-23 20:34:50.847     10.65      1
5878  3848003084          9321753 2023-08-23 09:48:27.913    118.01      1
5879  2410002412          9268637 2023-07-28 12:03:43.043     21.63      2
5880  2410002412          9310290 2023-08-18 15:02:05.467      7.95      2
5881  2776005063          9337991 2023-08-29 08:00:30.150     38.02      1

       kmpt1   kmpt2  year  month  day             time        date  hour  \
0      0.000   0.491  2019      5   21  15:39:24.907000  2019-05-21    15
1      0.000   0.491  2022      2   23  12:28:34.740000  2022-02-23    12
2      0.000   0.491  2015     11   23  05:27:52.947000  2015-11-23     5
3      0.000   0.491  2017     12   17  10:35:32.080000  2017-12-17    10
4      0.000   0.491  2018      1   28  06:54:17.827000  2018-01-28     6
...      ...     ...   ...    ...  ...              ...         ...   ...
5877   9.471  14.109  2023      7   23  20:34:50.847000  2023-07-23    20
5878   9.471  14.109  2023      8   23  09:48:27.913000  2023-08-23     9
5879   3.249   3.832  2023      7   28  12:03:43.043000  2023-07-28    12
5880   3.249   3.832  2023      8   18  15:02:05.467000  2023-08-18    15
5881  57.161  61.651  2023      8   29  08:00:30.150000  2023-08-29     8

      day_of_week weekday_or_weekend  \
0               1            Weekday
1               2            Weekday
2               0            Weekday
3               6            Weekend
4               6            Weekend
...           ...                ...
5877            6            Weekend
```

**Figure 9-28: Transformed incident dataset for Antwerp as a dataframe.**

### *9.1.3.2.3 Data cleaning and filtering*

After having created supplementary fields based on the information available in our initial raw dataset, we performed filtering in order to limit our incident occurrences to the same time period when the loop detector dataset is available, specifically from 2022-10-27 until 2023-08-29. For this time period, in the following Figures you can see some insights drawn from the incident dataset.

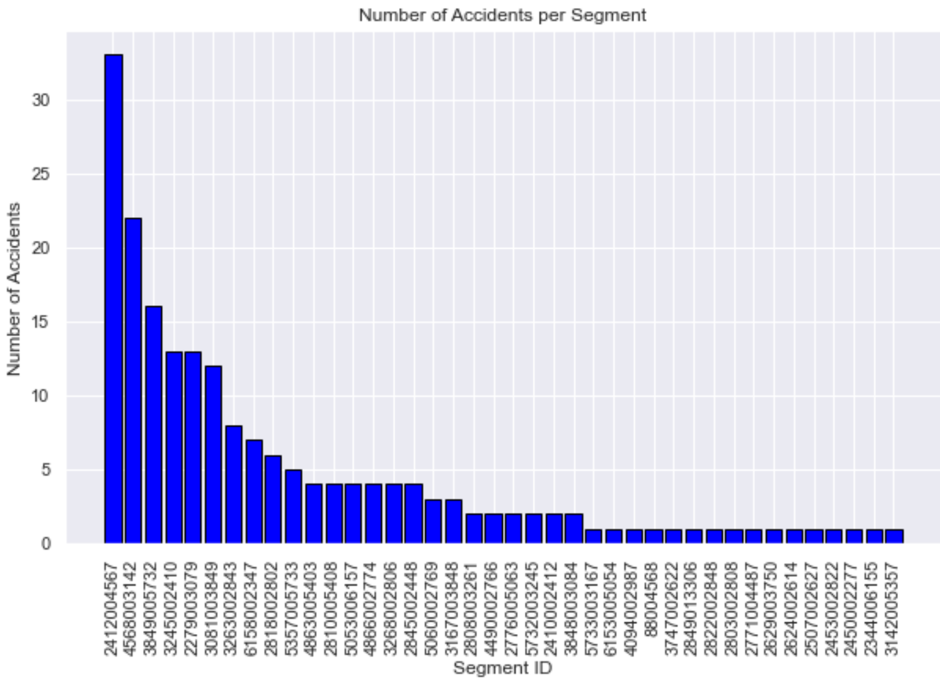



**Figure 9-29: Histogram depicting number of accidents per segment in Antwerp.**

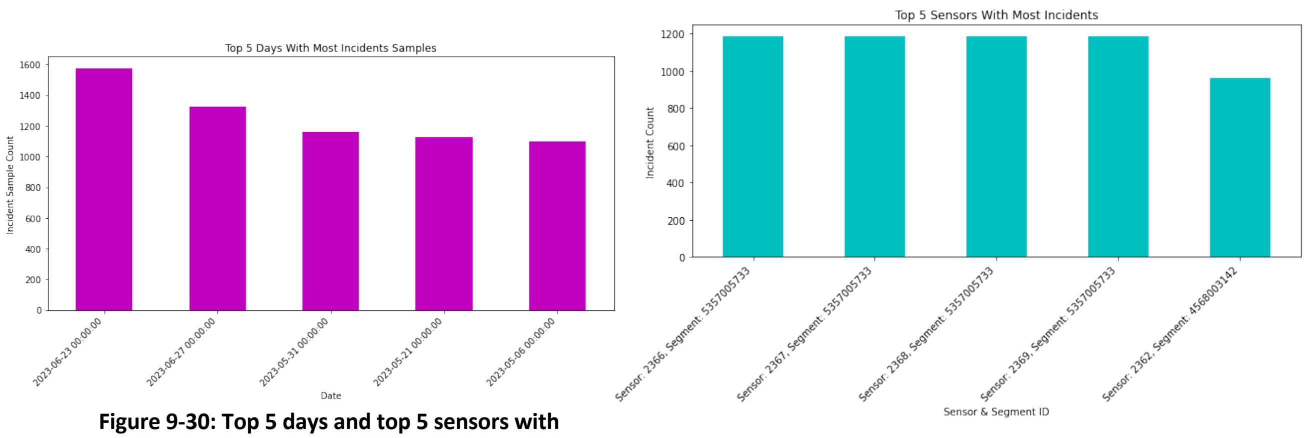



**Figure 9-30: Top 5 days and top 5 sensors with most recorded incidents respectively.**

### *9.1.3.2.4 Data transformation*

As specified in the previous section regarding the use case of Athens, similarly, after the data cleaning and the filtering, the dataset needed to be transformed in a way that it could be fed in the Machine Learning algorithms.

Based on the start timestamp and duration of the incidents specified in the relative columns of the dataset, we have created a transformer script which maps these timestamps to the corresponding start and end time. Finally, we were able to transform the filtered incidents dataset for directions 1 and 2 in 5-minute intervals where rows refer to timestamps and columns were the ids of the segments and the values of the matrix were 1 if this location and time corresponds to an incident occurrence, or 0 otherwise.

| | 497003192 | 2277002279 | 2279003079 | 2344006155 | 2410002412 | 2412004567 | 2448002450 | 2450002277 | 2616002344 | 2622002624 | 2624002614 | 26640037 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **2022-10-27 00:00:00** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| **2022-10-27 00:05:00** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| **2022-10-27 00:10:00** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| **2022-10-27 00:15:00** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| **2022-10-27 00:20:00** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| **2023-08-29 23:35:00** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| **2023-08-29 23:40:00** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

**Figure 9-31: Final incident dataset for Antwerp after necessary transformations.**

### *9.1.3.2.5 Limitations*

We would like to specify the fact that there are limitations to the presented steps, specifically regarding data filtering, along with inherent limitations stemming from the collected data itself, some of which are briefly discussed below:

- It is possible that some of the accidents are not registered, since from the input received from the stakeholder operators, the incidents have been recorded manually.
- Another possibility is that perhaps the recorded timings are not accurate. It is possible that an event which had happened on a timestamp *t*, is recorded on a later timestamp. However, this has a severe impact on the evaluation of our algorithms.

## 9.2 Data Preparation Process

The loop detector datasets need to be prepared in order to be fed to our methodology. First of all, data cleaning is performed, and the respective features are extracted. Moreover, the loop detector datasets for both case studies suffer from imbalance issues, as the majority of samples belong to the Normal class. The challenge of working with imbalanced datasets is that most machine learning techniques will ignore, and in turn have poor performance on, the minority class, although it is the performance on the minority class that is mostly important. To address this issue, the Synthetic Minority Oversampling Technique (SMOTE) (Chawla, Bowyer, Hall, & Kegelmeyer, 2002) and Tomek link (Tomek, 1976) is frequently employed. In our cases, we have chosen to combine SMOTE with Tomek links technique, as it has been shown that this method is much superior compared with that of using only one of the two (Zeng, Zou, Wei, Liu, & Wang, 2016) (Swana, Doorsamy, & Bokoro, 2022). Afterward, the data were normalized by the Robust Scaler, which scales the features using statistics that are robust to outliers. Table 9-7 summarizes the steps involved in the data-preprocessing process used in this research.

**Table 9-7: Preprocessing operations applied in the loop detector datasets.**

| Preprocessing Operation | Details |
|---|---|
| **Data cleaning** | Several filters were applied to:<br>• remove detectors which were not in the station aggregation file,<br>• flow reliability outliers,<br>• flow-occupancy-speed mismatches,<br>• detectors with more than 50% not a number entries (NaNs),<br>• stuck values (constant readings across time),<br>• isolated values,<br>• and atypical profiles.<br>Several types of imputation of missing/unreliable data were carried out on a portion of the readings, namely: polynomial, time k-nearest neighbor (KNN), free-flow speed imputation, spatial KNN, PPCA-based imputation, and weekday-based imputation. |
| **Resampling** | Resampling to 5-minute intervals. |
| **Features extraction** | The following features have been extracted for the classification task: Traffic_Variables*1, Upstream and downstream Traffic_Variables for adjacent detectors, Mean upstream and downstream Traffic_Variables |

| | |
|---|---|
| | of detector {5, 10, 15} minutes before, Mean upstream and downstream Traffic_Variables of detector {5, 10, 15} minutes after, Mean upstream and downstream Traffic_Variables of adjacent detectors {5, 10, 15} minutes before, Mean upstream and downstream Traffic_Variables of adjacent detectors {5, 10, 15} minutes after, time_of_day, day_of_week, is_weekend, is_holiday.<br>For the regression task, the same set of features have been extracted, with the exception of: Mean upstream and downstream Traffic_Variables of detector {5, 10, 15} minutes after and Mean upstream and downstream Traffic_Variables of adjacent detectors {5, 10, 15} minutes after.<br>*1{Flow} for Athens case study; {Flow, Occupancy, Speed} for Antwerp case study. |
| **Data Balancing *[2]** | SMOTE with Tomek Link.<br>*[2] Only used for the classification task |
| **Normalization** | Robust Scaler. |

Regarding the processing of the target dataset, the labelled incidents' dataset, as it has thoroughly been described per case study, we would like to acknowledge the existence of certain limitations in the steps outlined, particularly concerning data filtering, as well as inherent limitations associated with the collected data itself. Several factors contribute to these limitations, which are discussed herein. Firstly, some incidents may not have been captured and registered within the dataset. Although our analysis indicates that all incidents were recorded manually, with most being identified through CCTV cameras, the potential for incomplete incident registration remains. Secondly, there is a possibility of inaccurate timing in the recorded incidents. It is feasible that an event occurring at a specific timestamp could be recorded or logged at a later timestamp. Such inaccuracies have notable repercussions on the evaluation of our algorithms. Lastly, the filtering process we employed is not immune to errors. While the selection criteria were based on the expertise of stakeholders and dataset characteristics, there is a chance that some incidents with significant traffic implications may have been inadvertently overlooked and not accounted for in our analysis.

### *9.2.1 Further processing*

Apart from the steps of the Machine Learning pipeline, which has been presented in Chapter 8.1.1, we are presenting in detail some steps we have performed based on the nature of the data and the problem we are addressing.

***Over sampling***

Due to the scarcity of traffic data under incident conditions, the dataset of incident and non-incident conditions is imbalanced. To resolve the imbalanced dataset, an over-sampling strategy is performed to the incident dataset for the Supervised learning task. The over-sampling strategy balances the dataset by increasing the number of minority class samples. The synthetic minority oversampling technique (SMOTE) is a typical over sampling algorithm. For each instance in the minority class, the algorithm calculates the Euclidean distance between this instance and other instances and obtains its k-nearest neighbors. Then, the sampling ratio is chosen according to the imbalance ratio of the dataset. For each minority instance, several instances are selected from its k-nearest neighbors randomly. For this reason, we have utilized the Python imbalanced-learn package to run the SMOTE algorithm on the dataset, for the supervised learning task, for instance for the SVM classification of incident versus non-incident class.

***Data normalization***

In Neural Networks, the input vectors should be normalized before using them when the input vectors are large values, otherwise they cannot be categorized properly because of the properties of activation function. For SVM models, the normalization of input vectors is also required.

In the transport sector, Z-normalization and Minmax normalization are used. To reduce the influence of some extreme values, Z-score is used to transform traffic data. The Z-score normalizes traffic data by subtracting the mean and scaling to unit variance. In our case, we have used the scikit-learn python package, and specifically the MinMaxScaler(), to transform the values accordingly.

### *Feature Engineering*

In literature, it has been shown that finding the temporal correlations of traffic flow is essential when building a traffic incident detection model. Therefore, extracting the difference between normal traffic conditions and risky traffic conditions is critical. Moreover, knowing the spatial correlations of traffic flow is also important to the incident detection model. Based on shock wave theory, it can be inferred that some time must elapse for the influence of an incident to spread. Thus, traffic flow parameters obtained from adjacent upstream and downstream detectors should also be considered because traffic flow near an incident is more sensitive than is more distant traffic flow. The traffic flow parameters of upstream or downstream detectors change earlier; therefore, considering these variables can help the model detect incidents with less delay. (Li, Lin, Du, Yang, & Ran, 2022)

**Table 1.** Variables selected using the proposed temporal and spatial rules (Li et al. 2020).

| Variable | Name |
|---|---|
| Speed at the upstream detector just after the incident | s_up_0 |
| Volume at the upstream detector just after the incident | v_up_0 |
| Occupancy at the upstream detector just after the incident | o_up_0 |
| Speed at the downstream detector just after the incident | s_dn_0 |
| Volume at the downstream detector just after the incident | v_dn_0 |
| Occupancy at the downstream detector just after the incident | o_dn_0 |
| Speed difference between the upstream and downstream detectors just after the incident | s_up_dn |
| Volume difference between the upstream and downstream detectors just after the incident | v_up_dn |
| Difference in occupancy between the upstream and downstream detectors just after the incident | o_up_dn |
| Speed at the upstream detector t before the incident | s_up_t |
| Volume at the upstream detector t before the incident | v_up_t |
| Occupancy at the upstream detector t before the incident | o_up_t |
| Speed at the downstream detector t before the incident | s_dn_t |
| Volume at the downstream detector t before the incident | v_dn_t |
| Occupancy at the downstream detector t before the incident | o_dn_t |
| Mean upstream traffic speed during the 5 min before the incident | m_s_up |
| Mean downstream traffic speed during the 5 min before the incident | m_s_dn |
| Mean upstream traffic volume during the 5 min before the incident | m_v_up |
| Mean downstream traffic volume during the 5 min before the incident | m_v_dn |
| Mean upstream occupancy during the 5 min before the incident | m_o_up |
| Mean downstream occupancy during the 5 min before the incident | m_o_dn |
| Standard deviation of the upstream traffic speed during the 5 min before the incident | s_s_up |
| Standard deviation of the downstream traffic speed during the 5 min before the incident | s_s_dn |
| Standard deviation of the upstream traffic volume during the 5 min before the incident | s_v_up |
| Standard deviation of the downstream traffic volume during the 5 min before the incident | s_v_dn |
| Standard deviation of the upstream occupancy during the 5 min before the incident | s_o_up |
| Standard deviation of the downstream occupancy during the 5 min before the incident | s_o_dn |

Note: In the table, t equals: 30 s, 60 s, 90 s, 120 s, 150 s, 180 s, 210 s, 240 s, 270 s, 300 s.

**Figure 9-32: List of selected features, as described by (Li, Sheng, Du, Wang, & Ran., 2020).**

In the case of Athens, due to the limitations which have been discussed, we have selected the following features to be used as inputs in the Machine Learning algorithms:

- Flow
- Upstream and downstream flow for the adjacent detectors
- Mean upstream and downstream flow of the detector 5 minutes before
- Mean upstream and downstream flow of the detector 5 minutes after
- Mean upstream and downstream flow of the adjacent detectors 5 minutes before
- Mean upstream and downstream flow of the adjacent detectors 5 minutes after;

whereas for Antwerp we have also included data for speed and occupancy accordingly.

For the deep learning algorithms, we chose 5-time steps to make the sequences. Hence, it is going to look at the 25 minutes before each point to train the model. In the experiments, we select the traffic flow of the past 25 minutes, which is a time sequence of 5 data points.

## 9.3 Evaluation of traditional ML models for unplanned incidents

A thorough discussion of the results obtained in addition to a comparison of the performance of the employed algorithms is presented in Chapter 9.7.1.1.

### 9.3.1 Athens Case Study

In Table 9-8, the evaluation for the methods used for detecting unplanned events in Attiki Odos is illustrated.

**Table 9-8: Evaluation of methods for detecting unplanned events in Athens (Attiki Odos) dataset.**

| Algorithm | Precision | Recall | F1-Score |
|---|---|---|---|
| SVM (per timestamp) | 0.58 | 0.97 | 0.64 |

| | | | |
|---|---|---|---|
| Isolation Forest (per timestamp) | 0.012 | 0.44 | 0.023 |
| BCNN (per timestamp) | 0.012 | 0.94 | 0.025 |
| WNN (per timestamp) | 0.05 | 0.96 | 0.09 |
| Autoencoder (per timestamp) | 0.03 | 0.49 | 0.05 |
| Bidirectional LSTM (per timestamp) | 0.19 | 0.43 | 0.26 |
| Random Forest (per timestamp) | 0.95 | 0.64 | 0.71 |
| Graph Neural Network (per timestamp) | 0.66 | 0.48 | 0.555 |
| AIMSUN (per timestamp) | 0.08 | 0.50 | 0.14 |

### *9.3.2 Antwerp Case Study*

In Table 9-9, the evaluation for the methods used for detecting unplanned events in Antwerp E313 highway is illustrated. Aimsun's methodology could not be tested in this use case, as the model developed for the Antwerp use case was only suitable for running offline simulations (as specifically requested by user partners of this use case).

**Table 9-9: Evaluation of methods for detecting unplanned events in Antwerp (Highway E313) dataset.**

| Algorithm | Precision | Recall | F1-Score |
|---|---|---|---|
| SVM (per timestamp) | 0.62 | 0.96 | 0.753 |
| Isolation Forest (per timestamp) | 0.019 | 0.48 | 0.037 |
| BCNN (per timestamp) | 0.02 | 0.92 | 0.039 |
| WNN (per timestamp) | 0.07 | 0.94 | 0.130 |
| Autoencoder (per timestamp) | 0.03 | 0.48 | 0.056 |
| Bidirectional LSTM (per timestamp) | 0.23 | 0.47 | 0.301 |
| Random Forest (per timestamp) | 0.98 | 0.79 | 0.86 |
| Graph Neural Network (per timestamp) | 0.63 | 0.47 | 0.538 |

## *9.4 Evaluation of AutoML Models for Unplanned Incident Detection*

The measures used to monitor the performance are those as discussed in Chapter 8.1.8. More precisely, for the unplanned incident detection using our developed AutoML framework, the scoring function used to evaluate multiple machine-learning algorithms uses the following metrics: precision, recall, and F1 score.

A thorough discussion of the results obtained in addition to a comparison of the performance of the employed algorithms is presented in Chapter 9.7.1.2.

### *9.4.1 Athens Use Case*

For the use case of Athens, the results are depicted in Table 9-10. We have made the decision to compare the outcome of our methodology with an Unsupervised method (Isolation Forest) and a data-driven AIMSUN algorithm (Torrent-Fontbona F. , Dominguez, Fernandez, & Casas, 2023), already presented in the section 9.3.

**Table 9-10: Comparison of our approach with sampled baseline methods - Athens.**

| Algorithm | Precision | Recall | F1-score |
|---|---|---|---|
| AutoML (per timestamp) | 0.83 | 0.62 | 0.71 |
| Isolation Forest (per timestamp) | 0.012 | 0.44 | 0.023 |
| AIMSUN (per timestamp) | 0.08 | 0.50 | 0.14 |

The algorithm which has been selected as the optimal from our methodology is the following:

*LinearSVR(GradientBoostingRegressor(input_matrix, alpha=0.95, learning_rate=1.0, loss=huber, max_depth=1, max_features=0.6000000000000001, min_samples_leaf=15, min_samples_split=11, n_estimators=100, subsample=1.0), C=20.0, dual=False, epsilon=0.001, loss=squared_epsilon_insensitive, tol=0.0001)*

Regarding the two baseline approaches, Isolation Forests are generally used in an unsupervised manner and only require a few conditions to separate anomalies from normal observations when compared to other methods which use basic distance and density measures. There are several works in the field of AID which use Isolation Forests, including (Zhu, Wang, Yan, Guo, & Tian, 2022). Their low linear time complexity and small memory requirements aid in eliminating major computational cost of distance calculation in all distance and density-based methods and can perform well in a multi-dimensional feature space. For Aimsun's baseline and the incident detection approach, we invite the reader to consult Chapter 5.2.1.8.

### *9.4.2 Antwerp Use Case*

For the use case of Antwerp, the results are depicted in Table 9-11. A decision to compare the outcome of our methodology with a Supervised method (Support Vector Machine) and a Generative Neural Network (AutoEncoder), already presented as part of section 9.3

**Table 9-11: Comparison of our approach with sampled baseline ML methods - Antwerp.**

| Algorithm | Precision | Recall | F1-score |
|---|---|---|---|
| AutoML (per timestamp) | 0.77 | 0.52 | 0.54 |
| SVM (per timestamp) | 0.62 | 0.96 | 0.753 |
| Autoencoder (per timestamp) | 0.03 | 0.48 | 0.056 |

The pipeline which has been selected as the optimal from our methodology is the following:

```
Pipeline(steps=[('stackingestimator',
          StackingEstimator(estimator=GaussianNB())),
         ('decisiontreeclassifier',
          DecisionTreeClassifier(criterion='entropy', max_depth=9,
                      min_samples_leaf=3, min_samples_split=7,
                      random_state=42))])
```

Regarding the two baseline approaches, support vector machine (SVM) is a supervised approach which is constructed from a unique learning algorithm that extracts training vectors that lie closest to the class boundary and makes use of them to construct a decision boundary that optimally separates the different classes of data ) (Cortes & Vapnik, 1995). Results from various studies have shown that SVM offers a lower misclassification rate, higher correct detection rate, lower false alarm rate and slightly faster detection time than other models in traffic incident detection (Yuan & Cheu, 2003). An Autoencoder is a generative unsupervised deep learning algorithm used for reconstructing high-dimensional input data using a neural network with a narrow bottleneck layer in the middle which contains the latent representation of the input data and have been used for Anomaly Detection tasks(for instance (Kopčan, Škvarek, & Klimo, 2021), (Ashraf, et al., 2020)), by comparing the output from a Decoder and the input to the Network and using a threshold, either manually set or learnt from the data itself. If the loss value exceeds the threshold, then the instance is categorized or classified as an anomaly.

## *9.5 Evaluation of advanced analytics-driven methodology for planned incidents*

A thorough discussion of the results obtained in addition to a comparison of the performance of the employed algorithms is presented in Chapter 9.7.2.

In Figure 9-33, flow data for few selected detectors over the period of four days for the use case of Athens and the speed, occupancy and flow for two days are depicted in respectively, for the use case of Antwerp are illustrated in Figure 9-34. From the plots, we can observe the following:

- There are apparent daily patterns in the measurement data, with peaks corresponding to what might be morning and evening rush hours.
- Traffic values tend to be lower during the nighttime hours and higher during the day, as it is expected.
- The highest variability in patterns is depicted in the occupancy dataset.

- Different detectors show varying patterns, suggesting differences in traffic behavior at these locations.

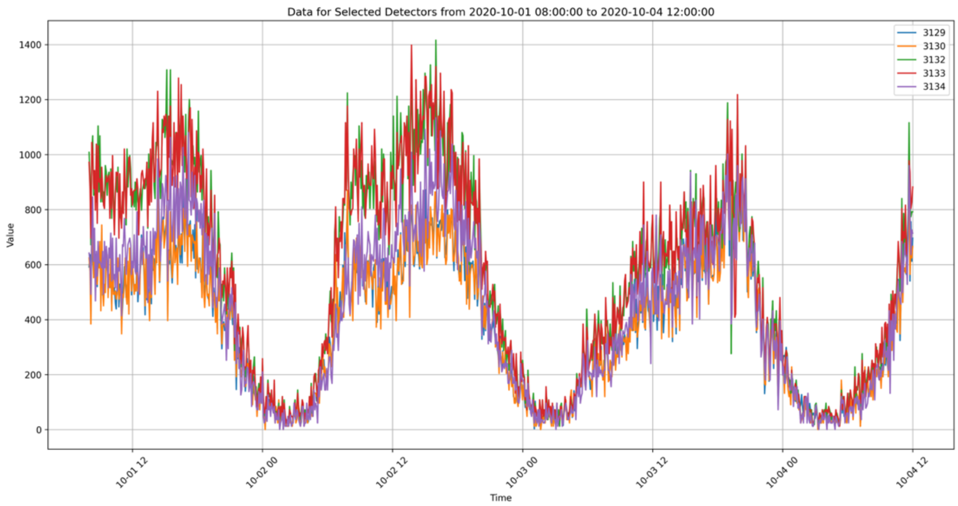



**Figure 9-33: Traffic flow for Athens' data for a specific loop detector over a period of four days.**

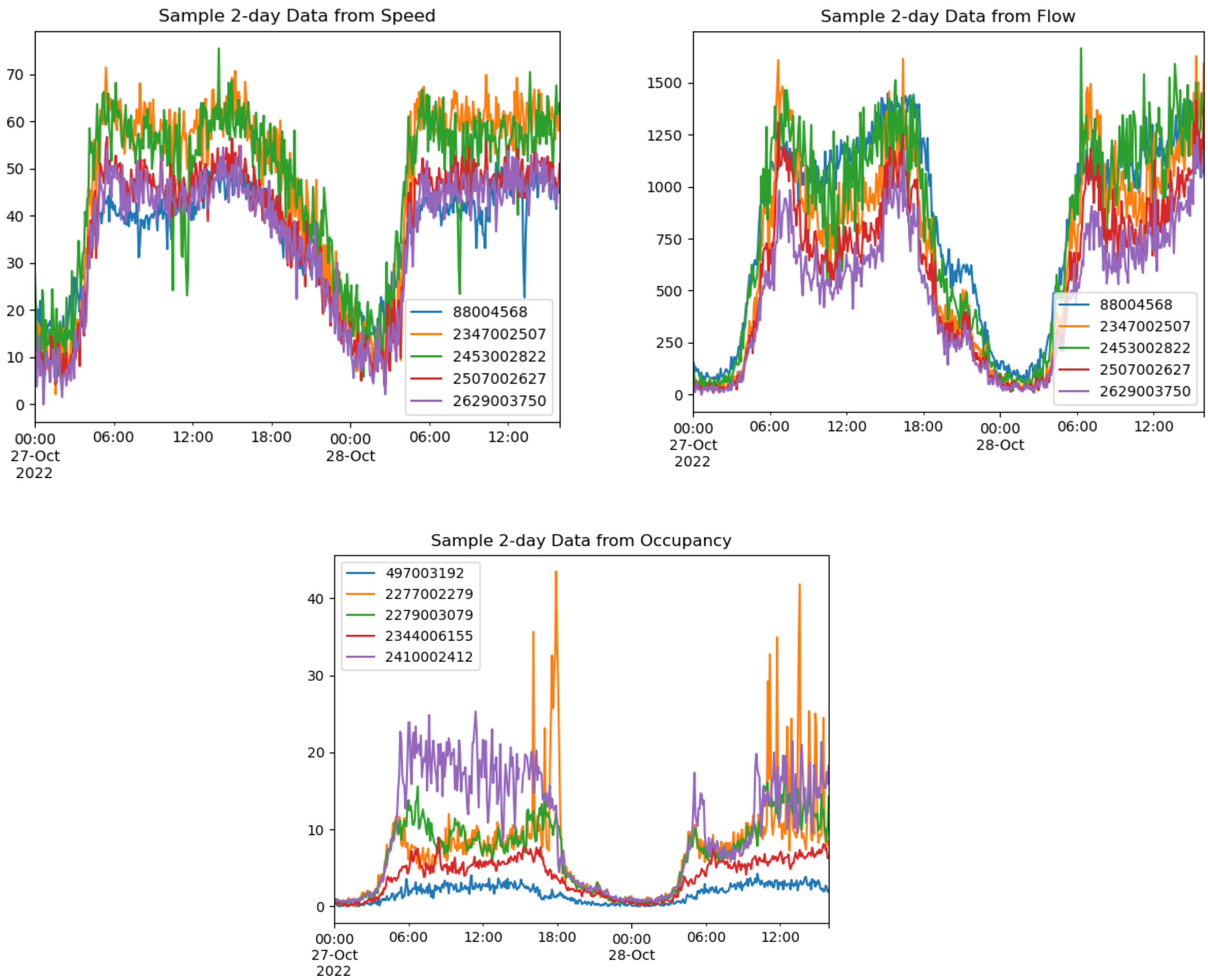
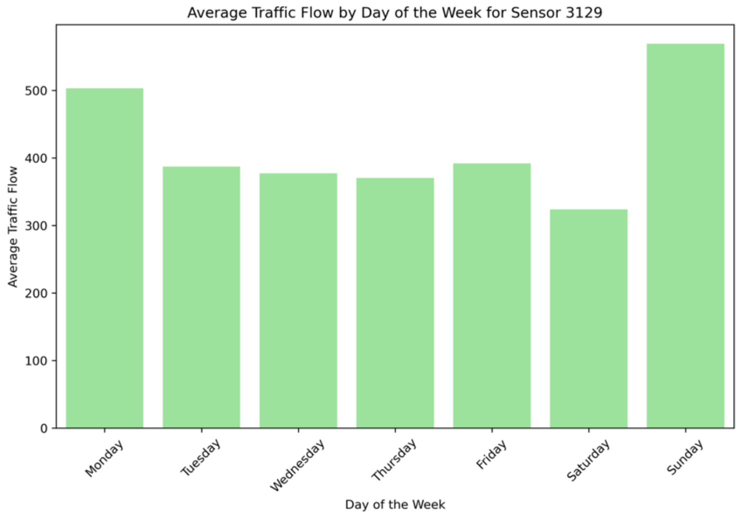



**Figure 9-34: Speed, flow and occupancy over selected segments over two days for Antwerp.**

As mentioned previously in Chapter 5, using histograms can provide insights into the distribution of traffic volumes or speeds. Figure 9-35 depict the average traffic flow by day of the week and by hour for a specific sensor in Athens dataset, whereas Figure 9-36 illustrates the average vehicle flow rate for all the sensors by hour for the Antwerp dataset. The conclusion we can draw is that there are spikes in the morning (between 8 and 10am) and in the afternoon (5-7pm) for the Athens use case, whereas for the Antwerp case there is significant drop across all traffic observations between 18:00 and 07:00.



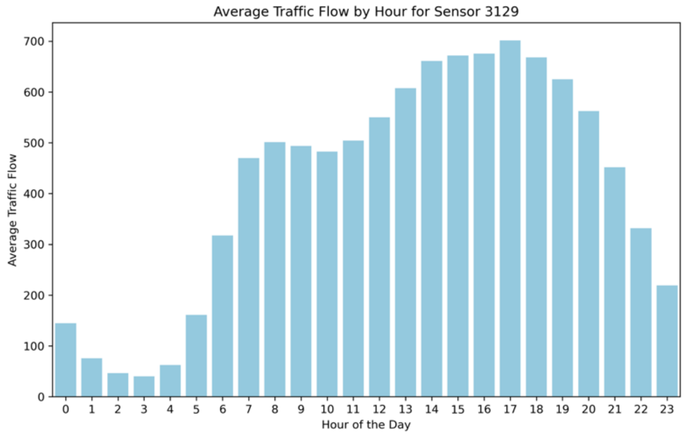




**Figure 9-35: Average traffic flow by day and by hour for Athens dataset.**

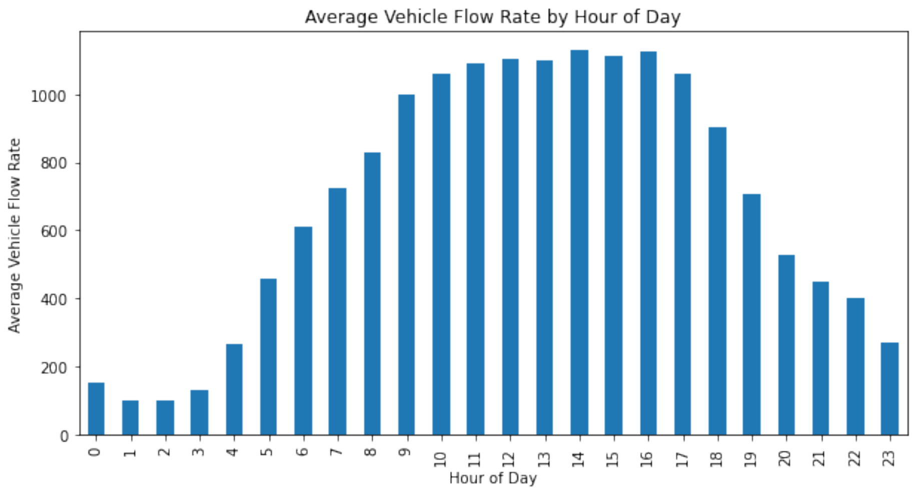



**Figure 9-36: Average traffic flow by hour for Antwerp dataset.**

In the following figures, we see two different types of visualisations of heatmaps. For the Athens use case, we analyse the traffic flow and depict the values in a matrix where x is the day of week and y the time of day, whereas for Antwerp for a selected segment, we use the dates as y and the daytimes as x values. We can deduce that weekdays have different traffic patterns compared to weekends and rush hours are identified morning and evening rush hours on weekdays and late afternoon on Sundays. Moreover, for Antwerp data, we identify the rush hour in the early afternoon between 15:00 and 18:00 for the specific segment.

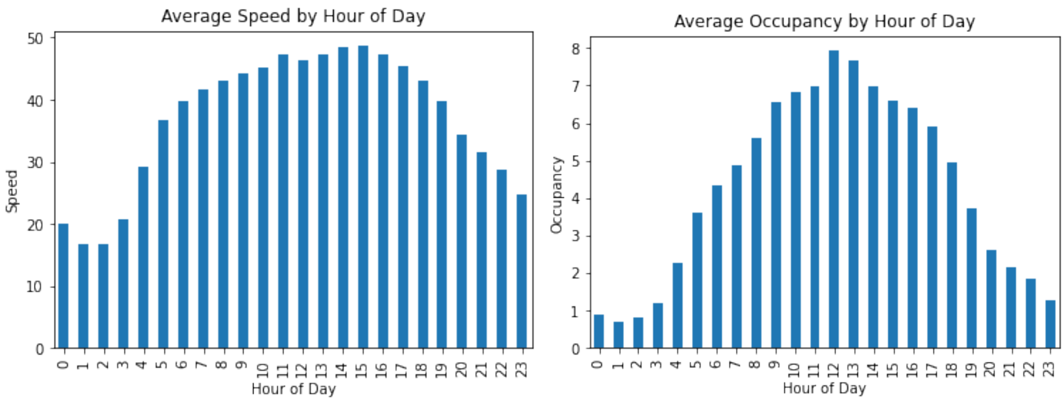



**Figure 9-37: Average Speed and Occupancy by hour of day - Antwerp.**

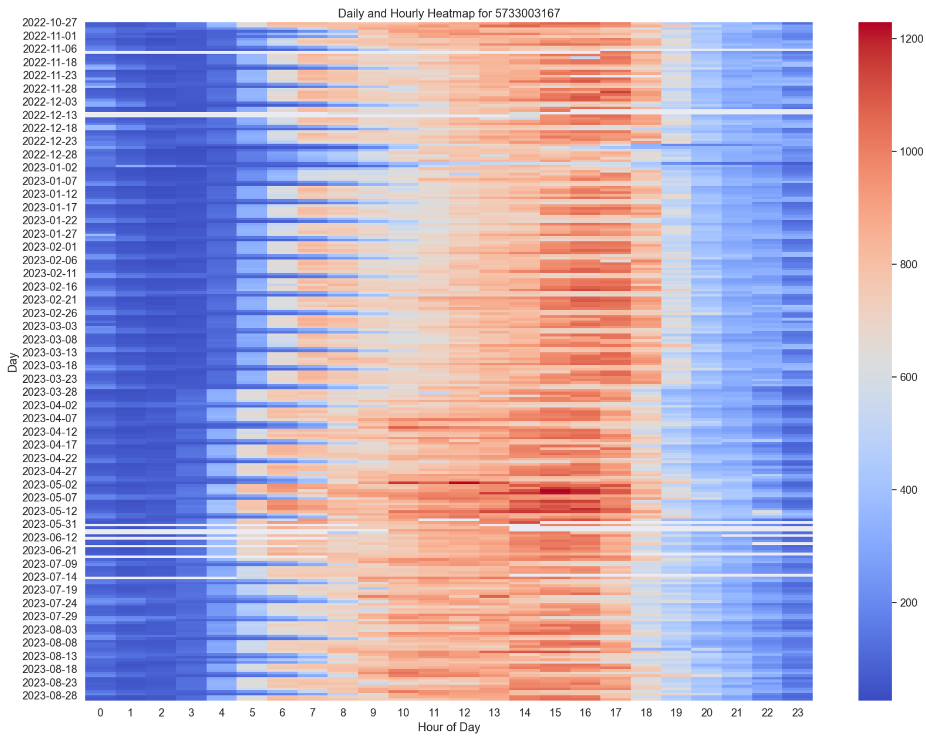



**Figure 9-38: Heatmap of flow based on time and date - Antwerp.**

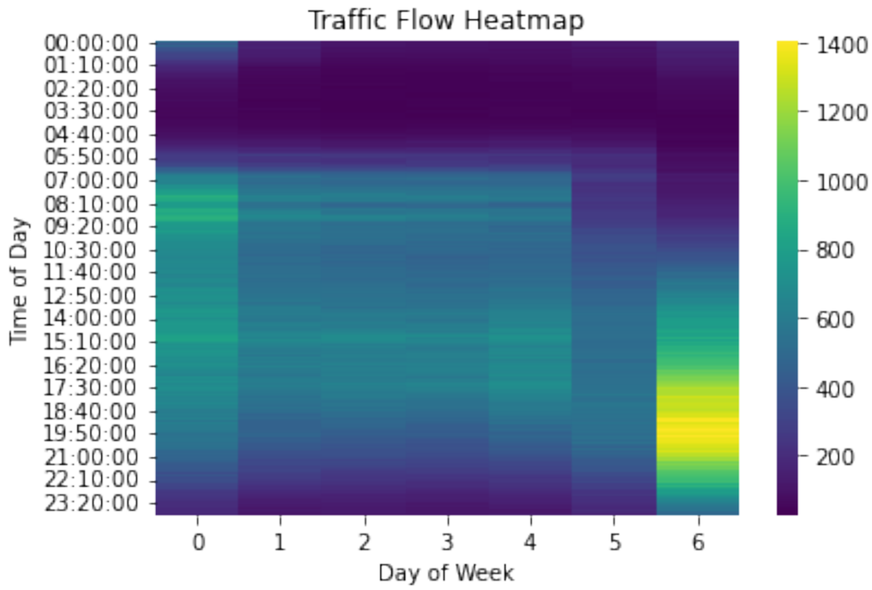



**Figure 9-39: Heatmap of flow based on time and day of week - Athens.**

In Figure 9-40 and Figure 9-41, we see two different types of visualisations of box plots. For the Athens use case, we analyse the traffic flow and depict the values of a specific sensor, whereas for Antwerp for all segments. By plotting box plots for traffic volumes or speeds for different times of the day or days of the week, one can identify variability and potential outliers. Periods with lower median speeds and high variability might be indicative of congestion. The insights drawn are in line with our previous analysis for rush hours, in addition to the fact that we notice quite a lot of outliers, especially for the Athens case. This may be due to non-recurrent unplanned incidents on specific dates during the analysed period.

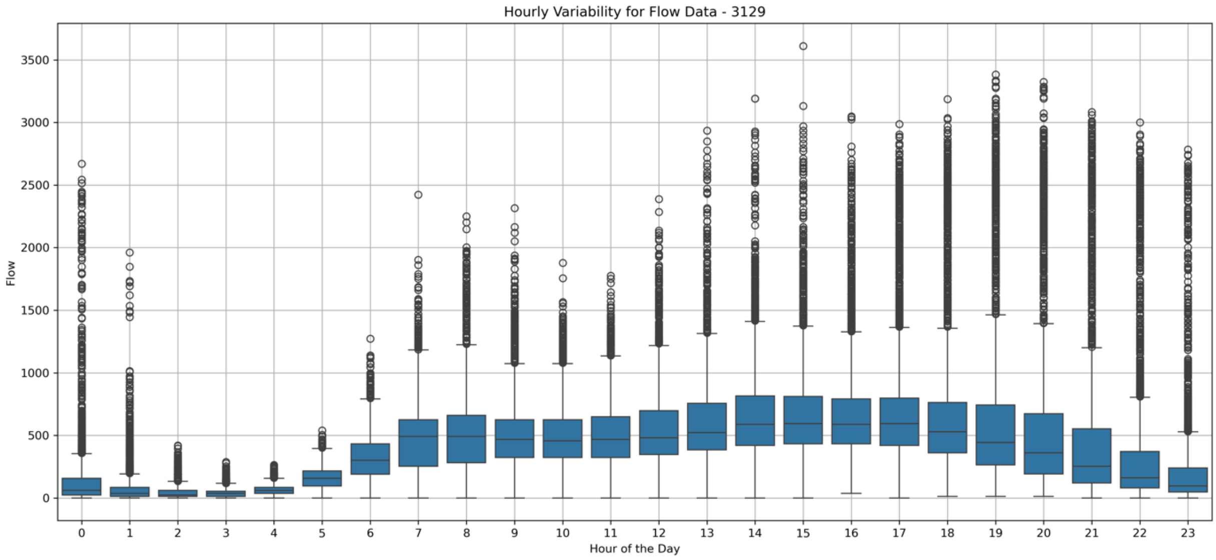



**Figure 9-40: Box plot depicting hourly variability for flow data for specific sensor - Athens.**

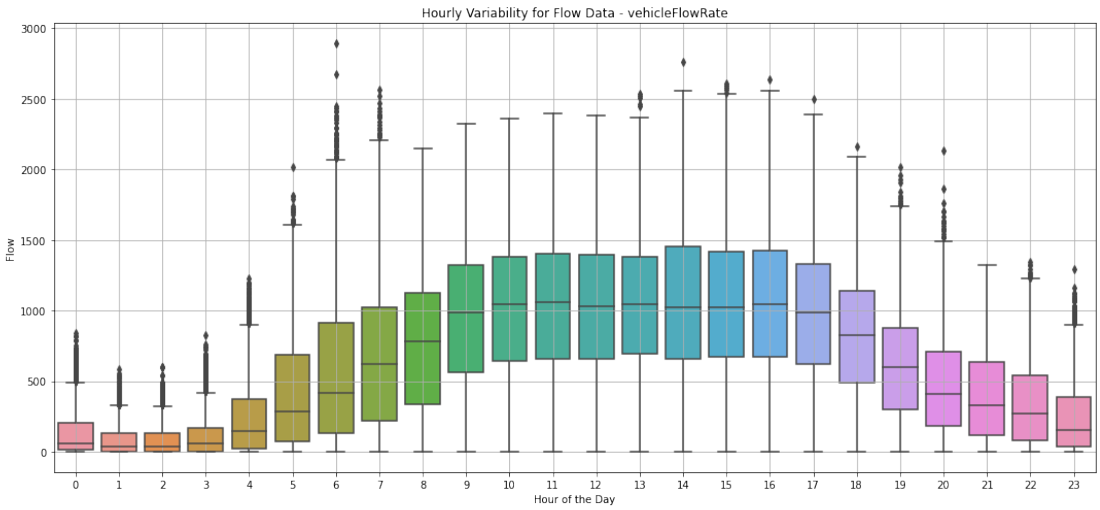



**Figure 9-41: Box plot depicting hourly variability for all segments - Antwerp.**

For Antwerp, we include a very rough analysis of those correlations indicating positive correlations between all three traffic measurements as shown in Figure 9-42. More fine-grained analysis can be made to deduce daily or weekly patterns and identify congestion.

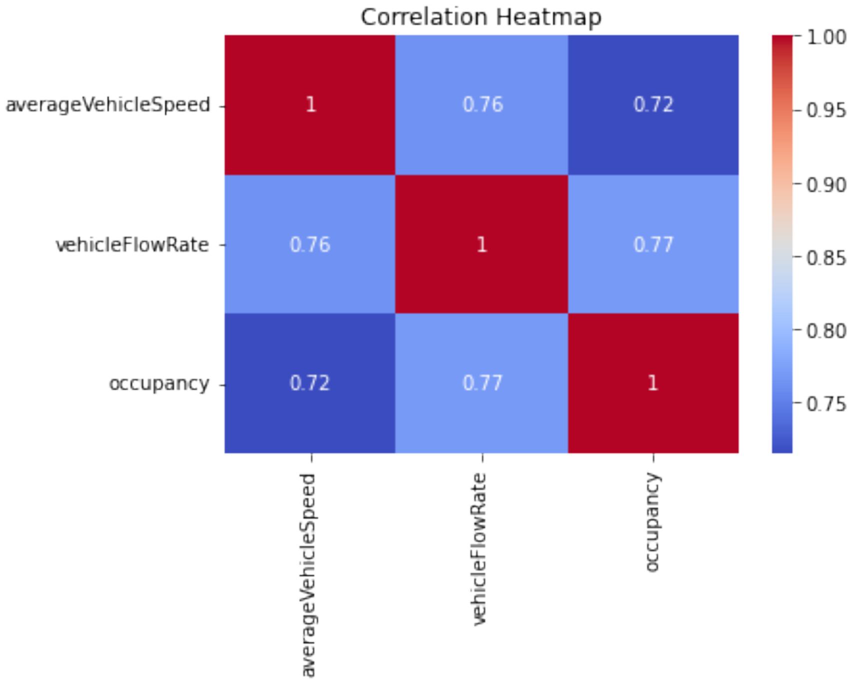



**Figure 9-42: Correlation heatmap between all traffic observations for Antwerp use case.**

By plotting two variables against each other, like traffic volume and speed, one can visually identify patterns or relationships as part of the depicted scatter plot. A downward trend in such a plot might indicate that as traffic volume increases, speeds decrease, signalling congestion. For Athens, as explained above, this analysis is not applicable. For Antwerp, such a coarse analysis is illustrated in Figure 9-43, where each colour represents a different traffic segment, and in certain cases we can deduce that in general high occupancy and low speed as shown in the left upper corner indicate congestion. The rest of the scatter plots for Antwerp (occupancy vs flow and speed vs flow) can be found in the below figures ( Figure 9-43 and Figure 9-44 ).

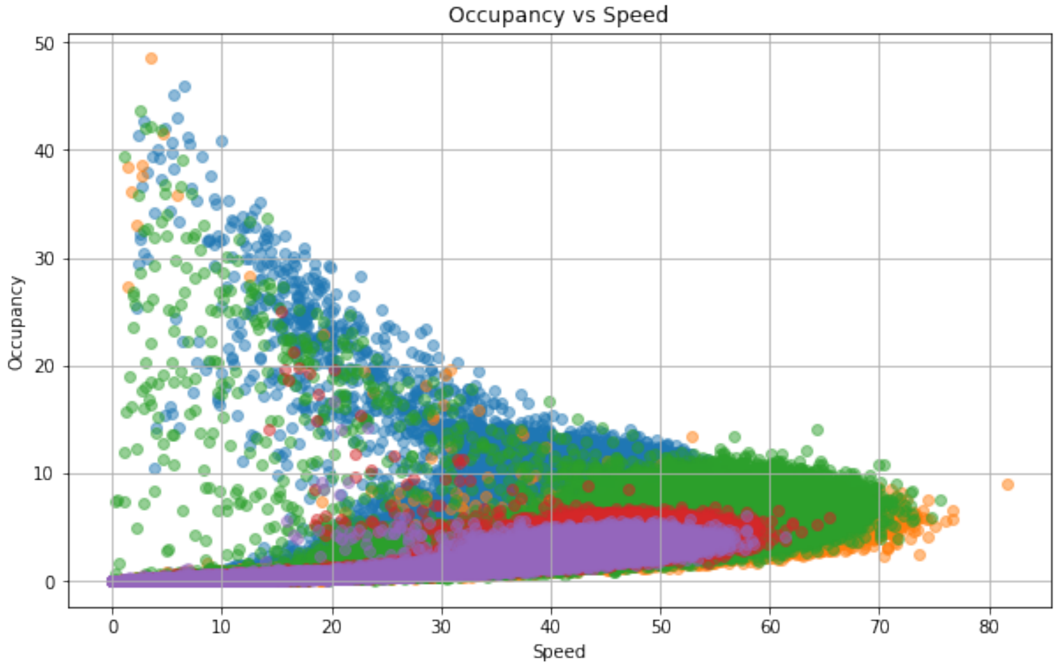



**Figure 9-43: Scatter plot for occupancy and speed - Antwerp.**

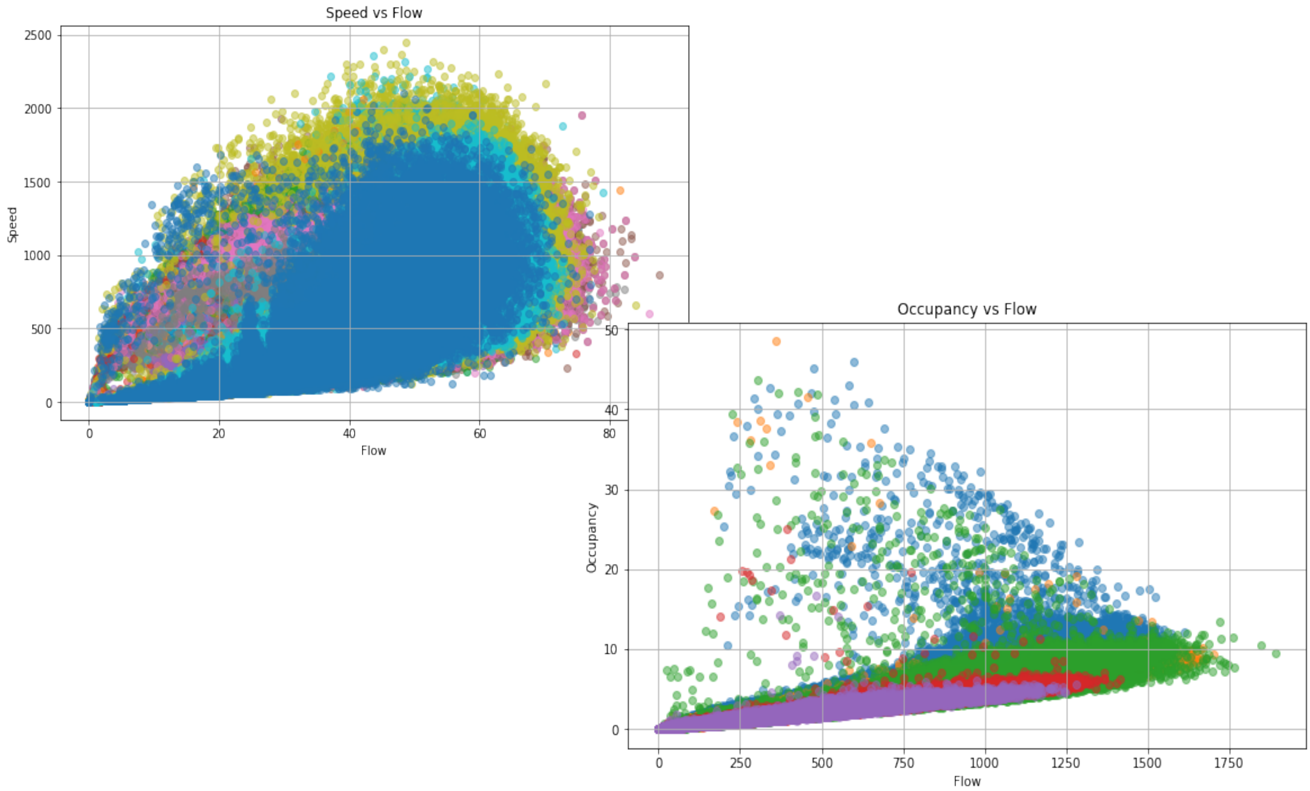



**Figure 9-44: Scatter plot flow vs speed and occupancy - Antwerp.**

The STL decomposition and rolling mean and standard deviation analysis have been performed for the Athens and Antwerp datasets and are illustrated in Figure 9-45 & Figure 9-46 and Figure 9-47 & Figure 9-48 respectively.

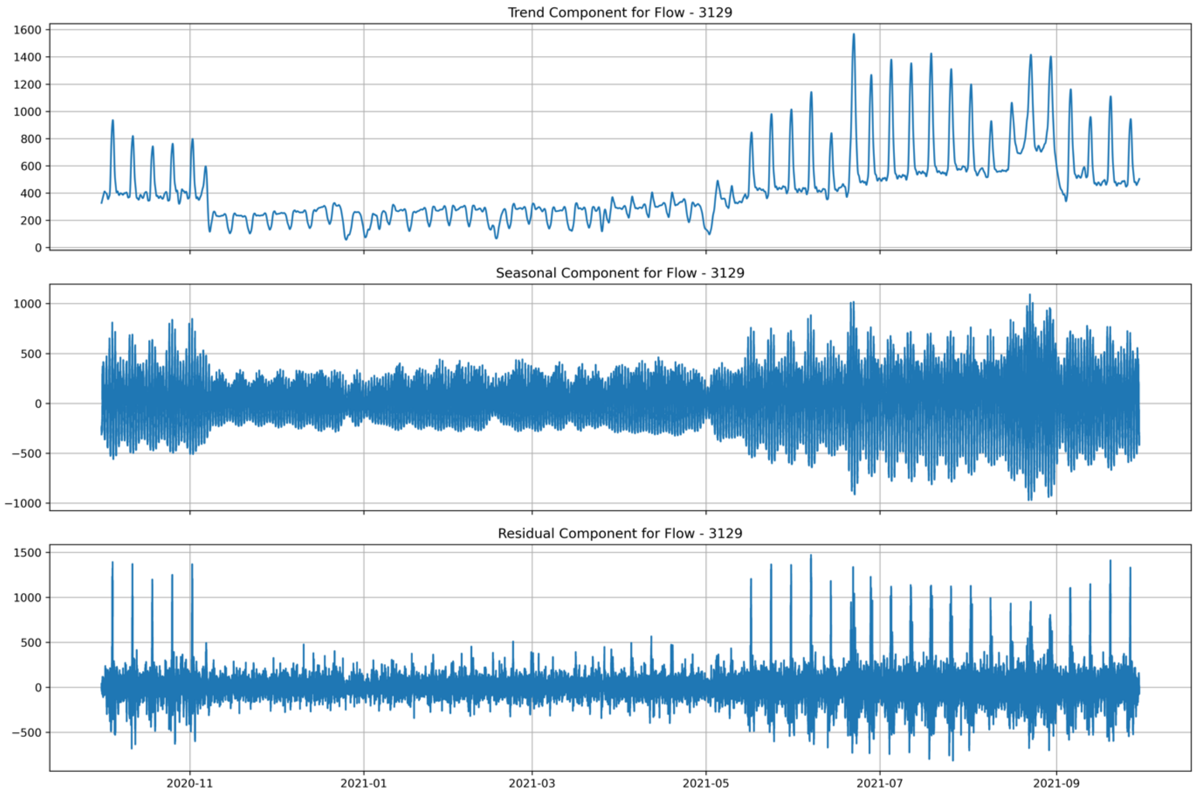



**Figure 9-45: STL decomposition - Athens.**

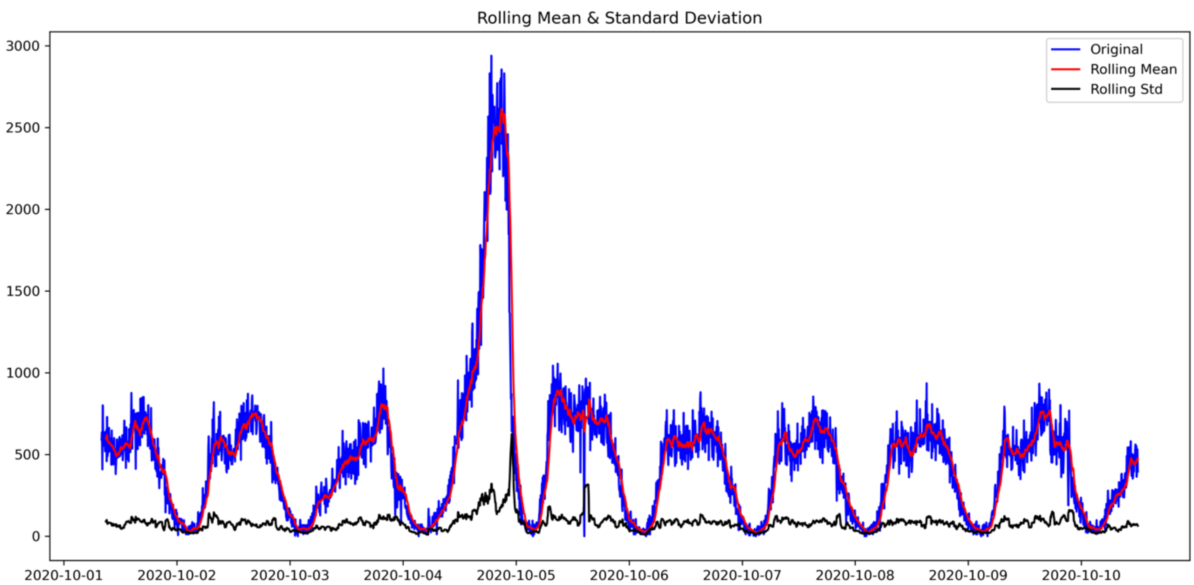



**Figure 9-46: Rolling mean and standard deviation - Athens.**

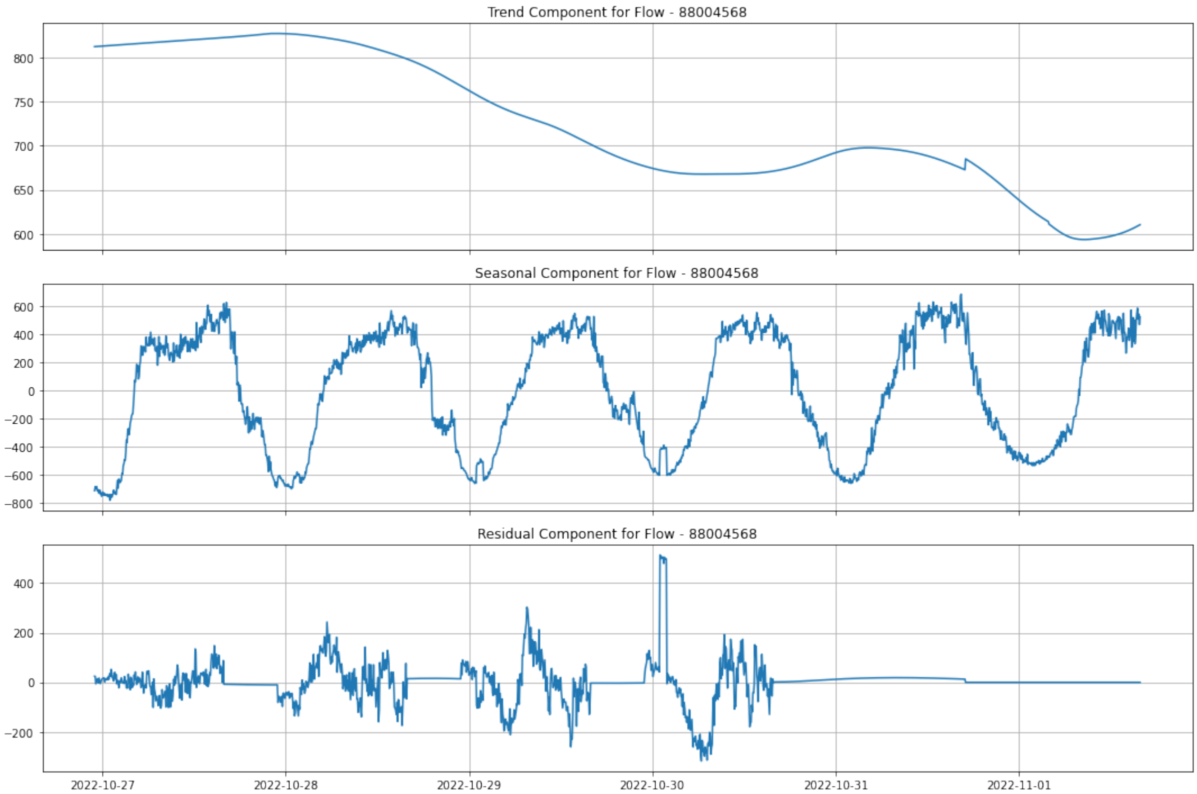



**Figure 9-47: STL decomposition - Antwerp.**

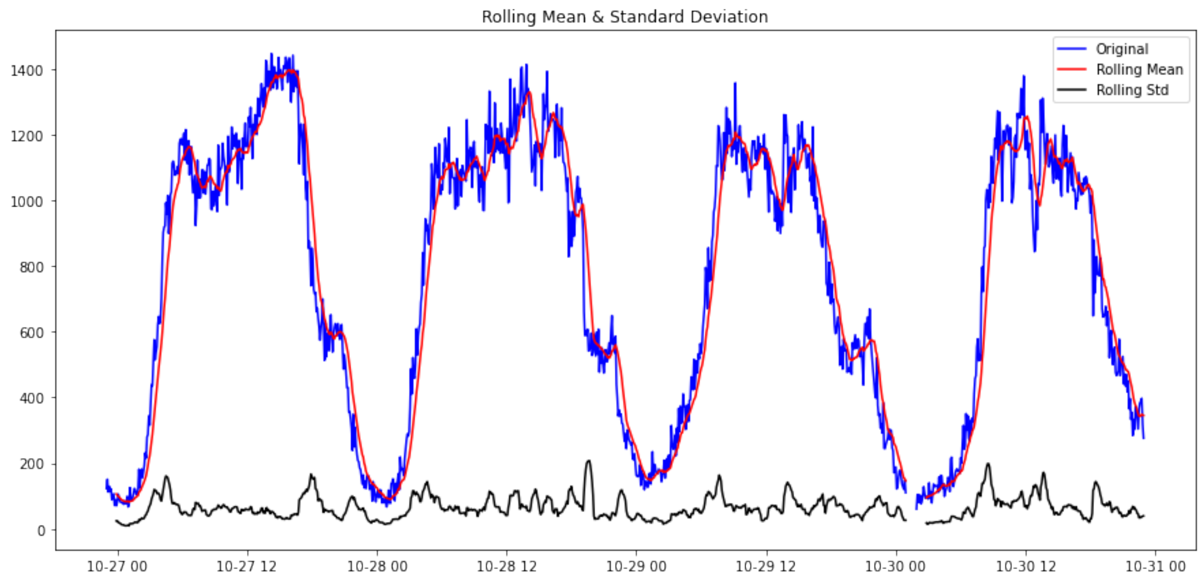



**Figure 9-48: Rolling mean and standard deviation - Antwerp.**

Moreover, as part of the analysis, to verify that we have managed to capture the time patterns correctly we predict the traffic flow using ARIMA models. These forecasts for both Athens and Antwerp case are shown in Figure 9-50 and Figure 9-53 respectively.

For Athens and Antwerp, the ACF and PACF are shown in the images below, in addition to the predictions using ARIMA models.

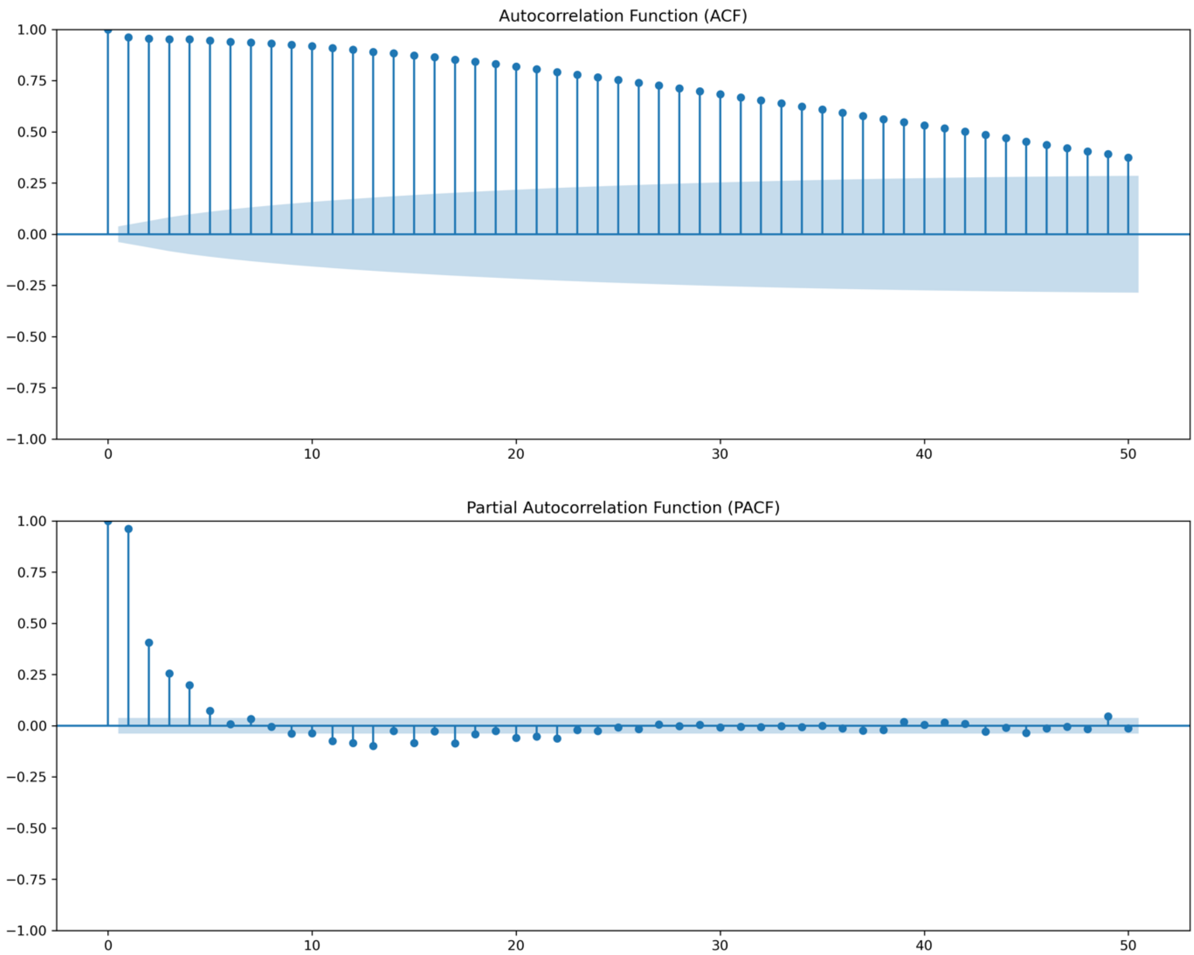



**Figure 9-49: ACF and PACF for Athens.**

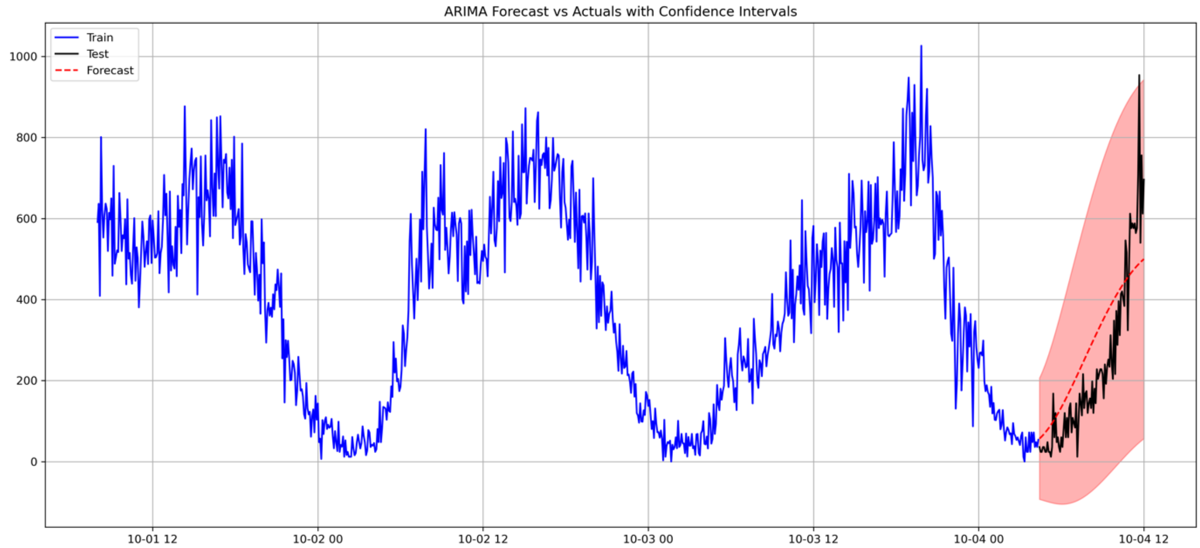



**Figure 9-50: Forecast using ARIMA for Athens use case.**

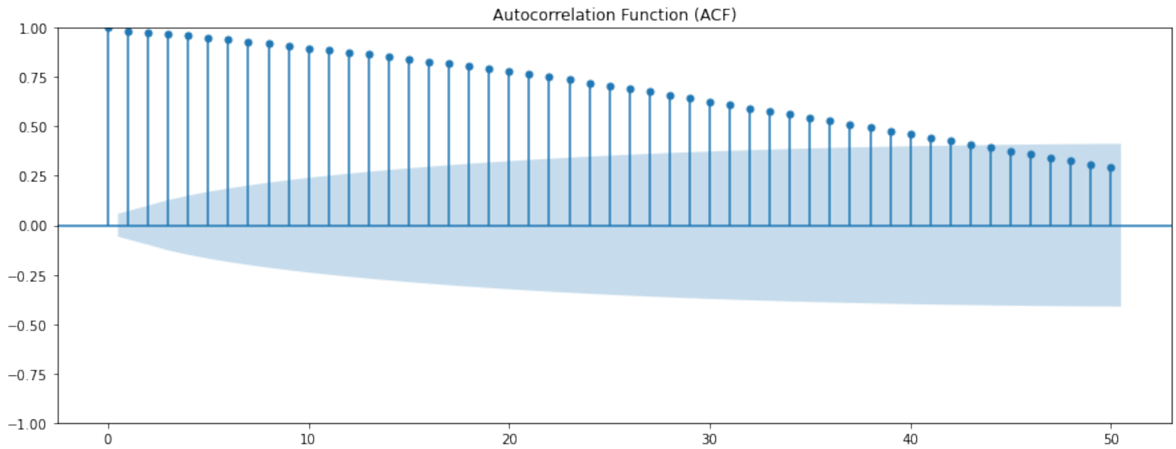



**Figure 9-51: ACF for Antwerp dataset.**

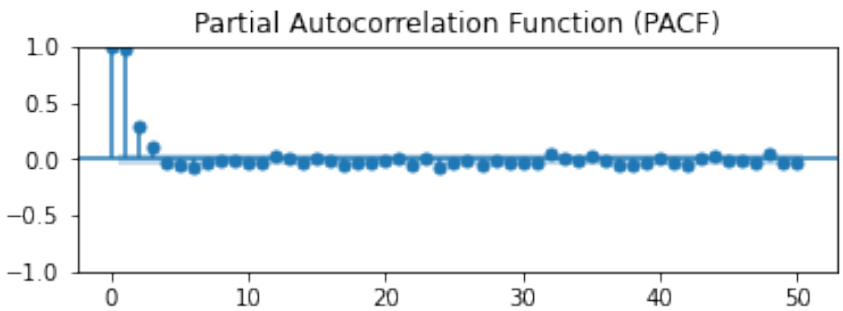



**Figure 9-52: PACF for Antwerp dataset.**

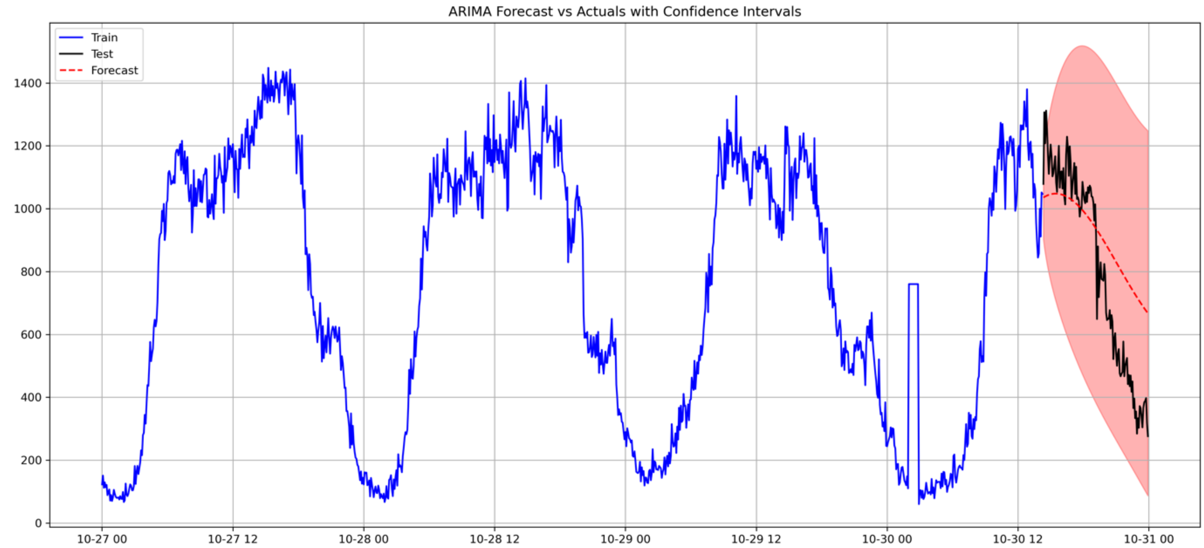



**Figure 9-53: Forecast using ARIMA for Antwerp use case.**

Regarding more advanced data-driven methods, we have selected to use and fine-tune a LSTM on the Athens and Antwerp datasets, for a longer and shorter period as test sets, to get an idea of how well the model has learnt the time patterns for a specific loop detector or segment. The predictions are illustrated in Figure 9-54 and Figure 9-55 respectively. As it is evident, the model has managed to capture the data in a very good manner, since it also includes as features the time of day, workday, day of week except for the traffic measurements.

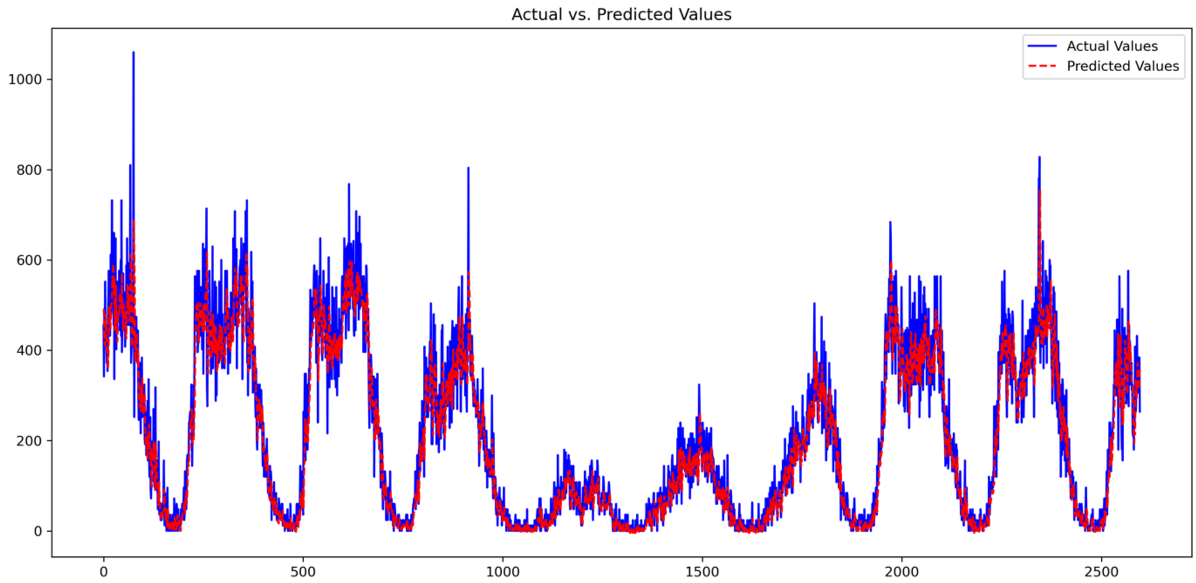



**Figure 9-54: Actual vs Predicted values for Athens using LSTM.**

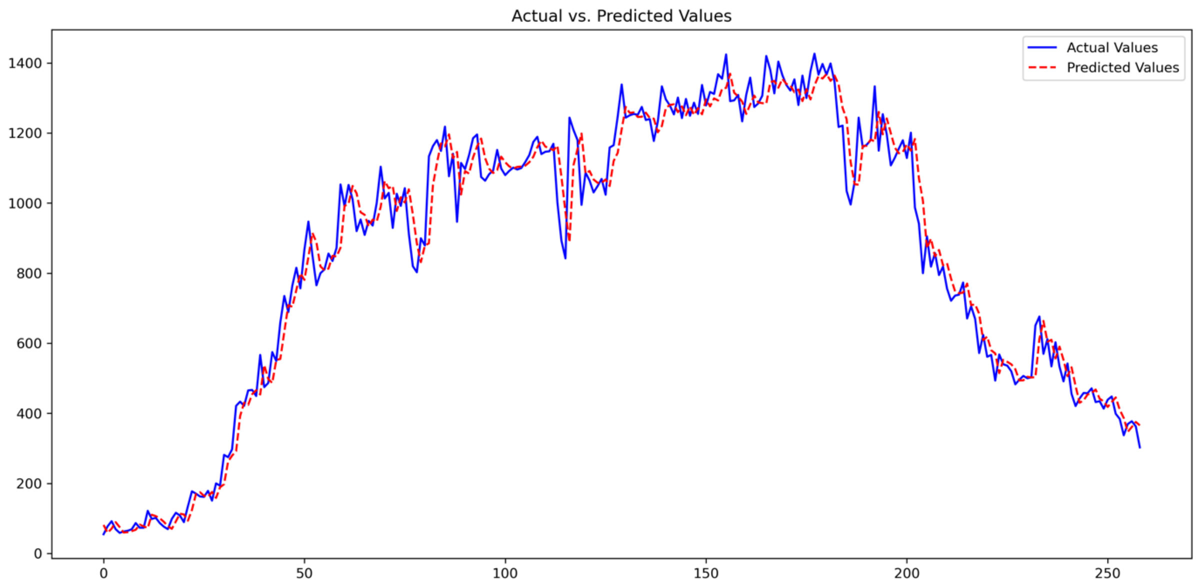



**Figure 9-55: Actual vs Predicted values for Antwerp using LSTM.**

Specifically, for our loop detector data we have employed k-means clustering. This algorithm partitions data into 'k' number of clusters. The elbow method is used to identify the optimal number of clusters and the k-means clustering is shown for Athens and Antwerp in the figures below (Figure 9-56 and Figure 9-57).

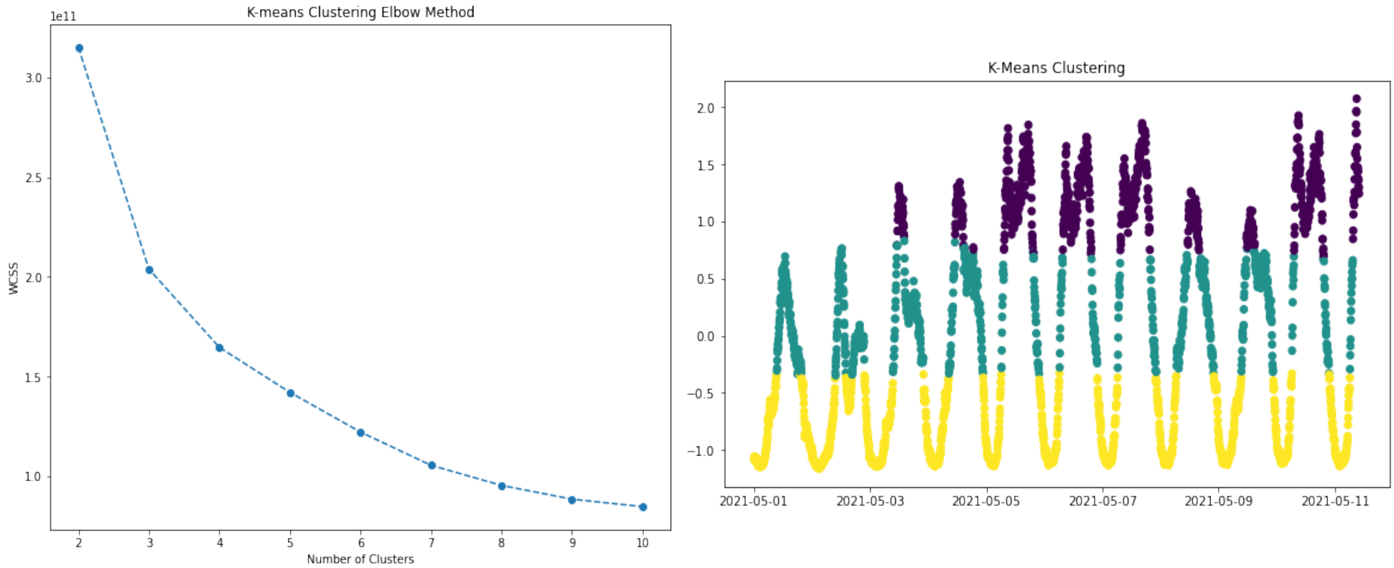



**Figure 9-56: K-means cluster number selection and clustering - Athens.**

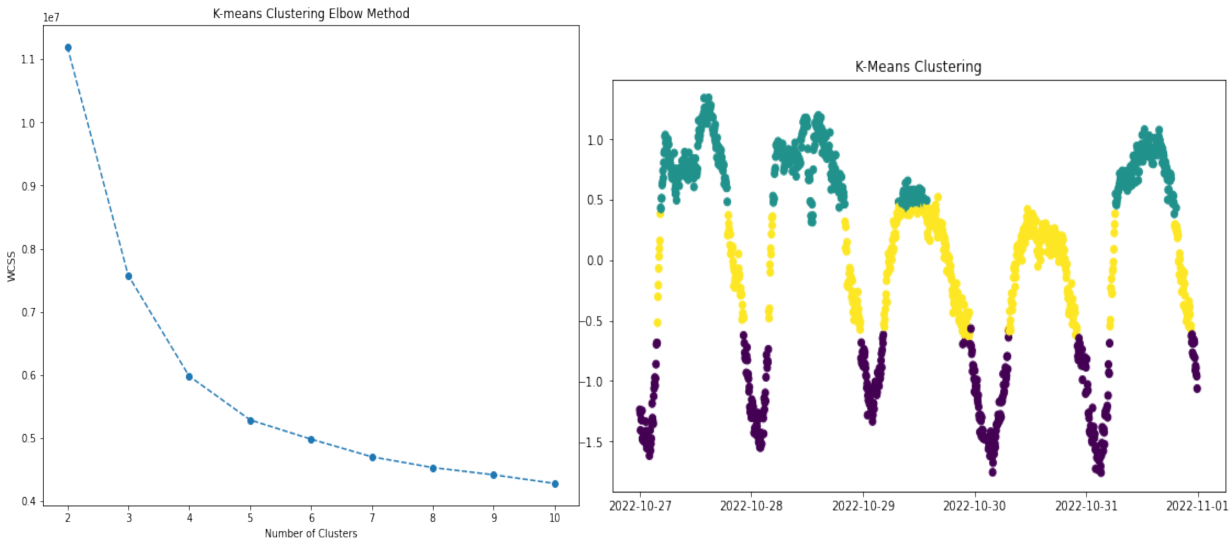



**Figure 9-57: K-means cluster number selection and clustering - Antwerp.**

## 9.6 Evaluation of Explainability and HITL Integration

### 9.6.1 Integration of Explainability features

When incidents like accidents, road blockages, or congestion occur, it's vital for operators and stakeholders to detect them quickly. However, merely identifying an incident isn't enough; understanding the "why" behind incident detection is crucial. This is where explainability comes in. An explainable system offers insights into the decision-making process, ensuring that human operators can trust the technology. Explainability ensures accountability, reduces false positives, and allows for better-informed interventions, all of which are pivotal in critical applications like transportation.

The concepts of LIME (Locally Interpretable Model-agnostic Explanations) and SHAP (SHapley Additive exPlanations) are central to the domain of explainable artificial intelligence (XAI), providing mechanisms to interpret the predictions made by machine learning models, as explained in the previous section. Thus, we have incorporated both LIME and SHAP libraries and performed various experiments, the results of which are detailed as part of this section.

The SHAP diagrams included in our analysis serve different purposes, each offering a unique perspective on model interpretability:

- **SHAP Waterfall Plot**: This plot shows how the features contribute to a single prediction for an individual instance. Starting from the base value (the mean prediction over the dataset), the waterfall plot sequentially adds or subtracts the SHAP values for each feature, illustrating how the model arrives at the final prediction. This plot is particularly useful for understanding the precise factors driving a specific prediction.

- **SHAP Dependence Plot**: This plot demonstrates the relationship between a specific feature's value and its SHAP value across all instances in the dataset. It helps in understanding the effect of that feature on the model's predictions and also shows interactions with other features, which are highlighted by colouring the points according to the value of another feature.

- **SHAP Force Plot**: The force plot provides a compact visualization of how different features push the prediction higher or lower relative to the base value. The plot visually represents the combined effects of the features as forces that either increase or decrease the final prediction.
- **SHAP Summary Plot**: This plot summarizes the impact of all features on the model's predictions across the entire dataset. It combines feature importance (how much each feature contributes overall) with feature effects (how the feature values affect the predictions), providing a holistic view of the model's behaviour.

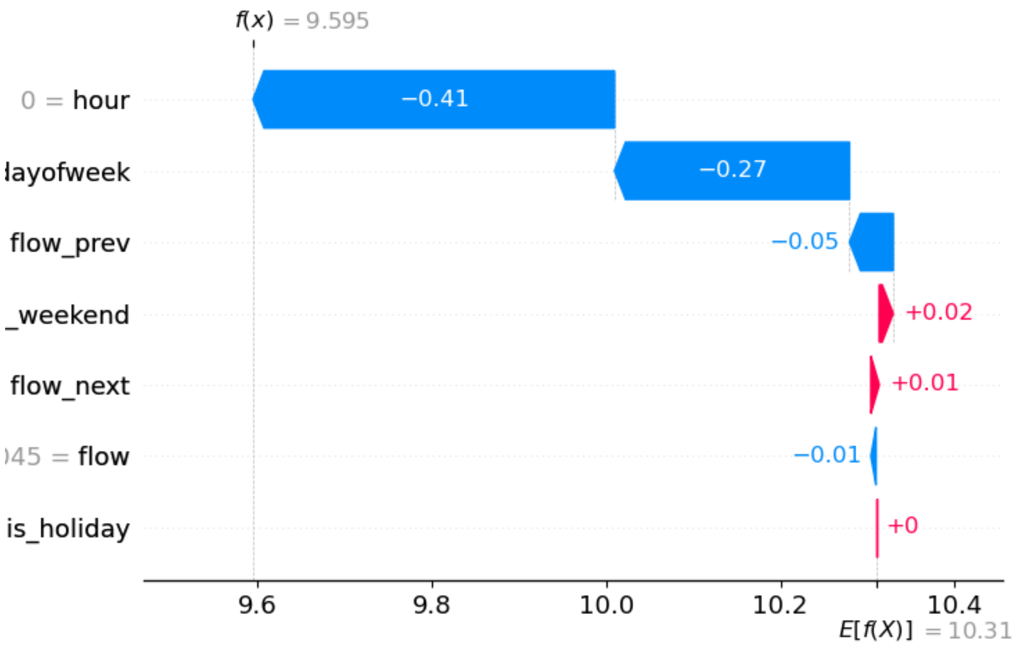



**Figure 9-58: SHAP Waterfall Plot for Athens' case study.**

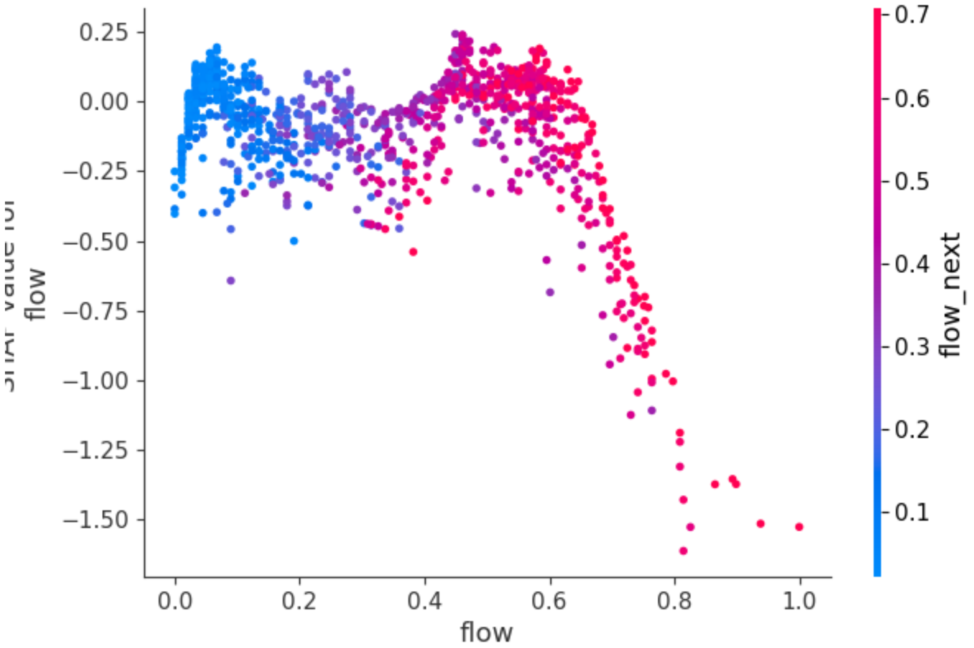



**Figure 9-59: SHAP Dependence Plot for Athens' case study.**

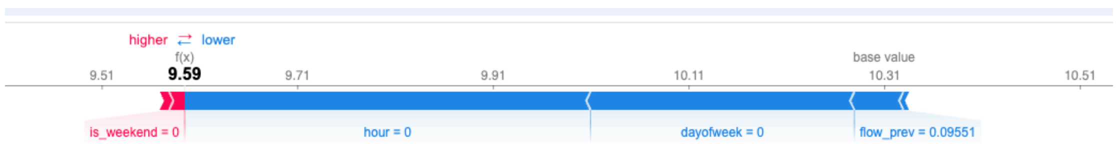



**Figure 9-60: SHAP Force Plot for Athens' case study.**

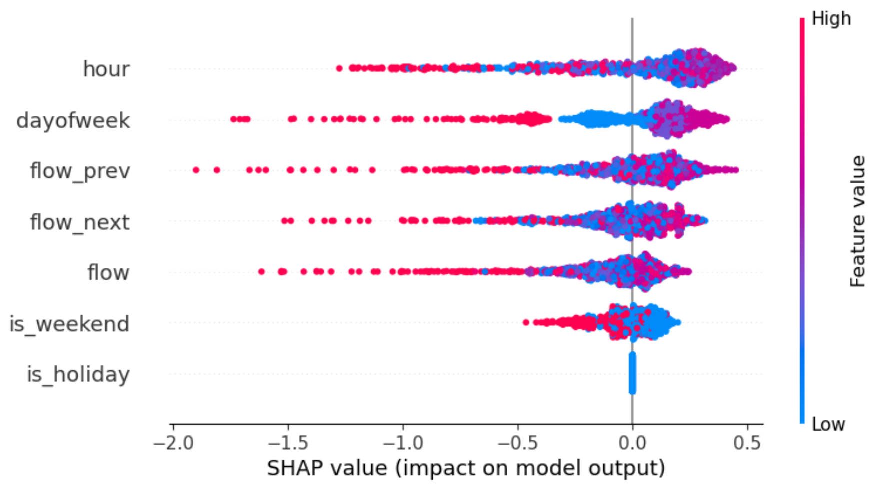



**Figure 9-61: SHAP Summary Plot for Athens' case study.**

The LIME diagrams included in our analysis help to break down and interpret the predictions made by the machine learning model on a granular level:

- **LIME Explanation Plot**: This plot visually represents the impact of individual features on a specific prediction. It shows which features contributed the most to the prediction, and whether they pushed the prediction higher or lower. The LIME explanation plot helps to identify the most influential features in a particular instance and provides a clear and interpretable summary of how the model made its decision.

- **LIME Explanation Heatmap**: This heatmap compares LIME explanations across multiple instances, allowing us to visualize how different features contribute to the model's predictions across various cases. Each row represents an instance, and each column represents a feature, with colour intensity indicating the magnitude of each feature's contribution. The heatmap helps in identifying patterns and consistencies in feature influences across multiple predictions, offering a broader view of the model's behaviour.

- **LIME Feature Importance Bar Plot**: This plot displays the impact of individual features on a specific prediction in the form of a horizontal bar chart. It clearly shows which features had the most significant influence on the prediction, indicating whether they pushed the prediction higher or lower. The bar plot helps identify the most critical features for a particular instance, providing a clear and interpretable summary of how the model arrived at its decision.

- **LIME Scatter Plot of Feature Influence**: This scatter plot illustrates the relationship between a specific feature's value and its influence on the model's prediction across all analysed instances. By plotting feature values against their corresponding LIME explanations, this chart helps in understanding the sensitivity of the model to changes in specific feature values and highlights any non-linear relationships.

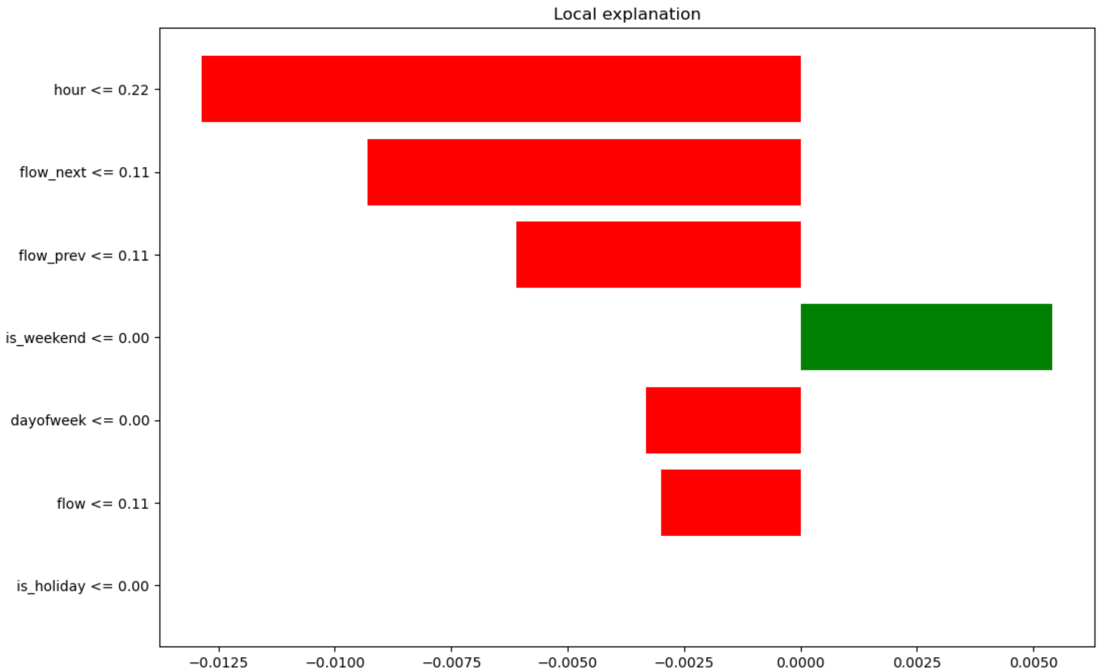
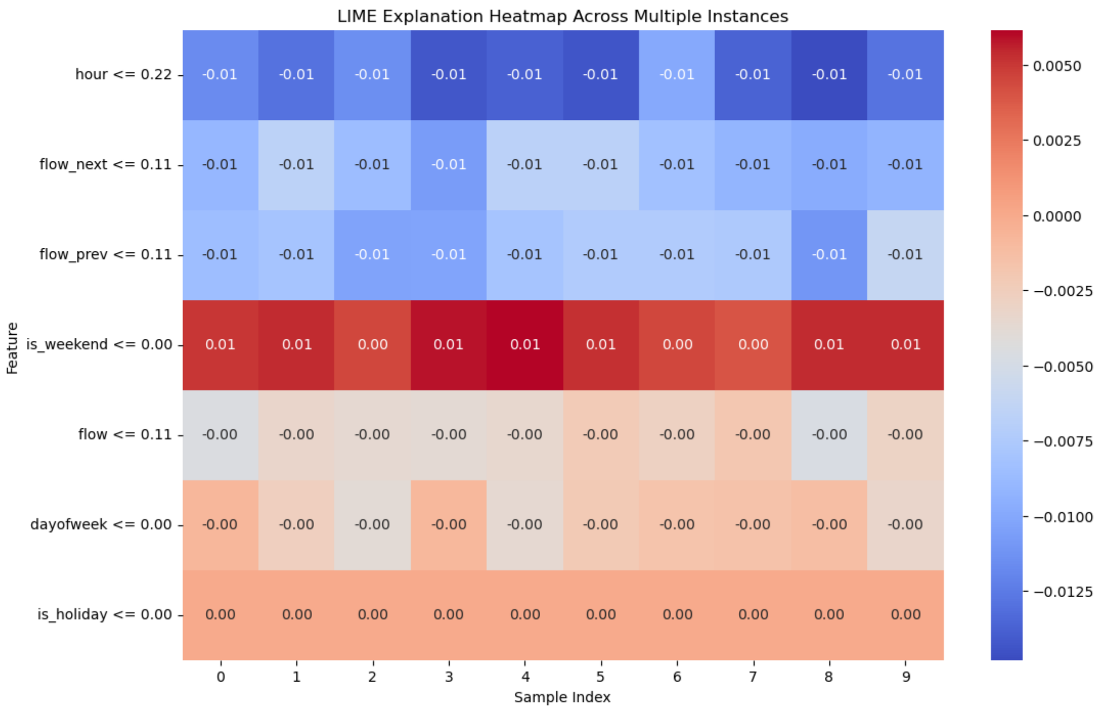



**Figure 9-62: LIME Explanation Plot for Athens' case study.**




| Feature | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| hour <= 0.22 | -0.01 | -0.01 | -0.01 | -0.01 | -0.01 | -0.01 | -0.01 | -0.01 | -0.01 | -0.01 |
| flow_next <= 0.11 | -0.01 | -0.01 | -0.01 | -0.01 | -0.01 | -0.01 | -0.01 | -0.01 | -0.01 | -0.01 |
| flow_prev <= 0.11 | -0.01 | -0.01 | -0.01 | -0.01 | -0.01 | -0.01 | -0.01 | -0.01 | -0.01 | -0.01 |
| is_weekend <= 0.00 | 0.01 | 0.01 | 0.00 | 0.01 | 0.01 | 0.01 | 0.00 | 0.00 | 0.01 | 0.01 |
| flow <= 0.11 | -0.00 | -0.00 | -0.00 | -0.00 | -0.00 | -0.00 | -0.00 | -0.00 | -0.00 | -0.00 |
| dayofweek <= 0.00 | -0.00 | -0.00 | -0.00 | -0.00 | -0.00 | -0.00 | -0.00 | -0.00 | -0.00 | -0.00 |
| is_holiday <= 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |

**Figure 9-63: LIME Explanation Heatmap Plot for Athens' case study.**

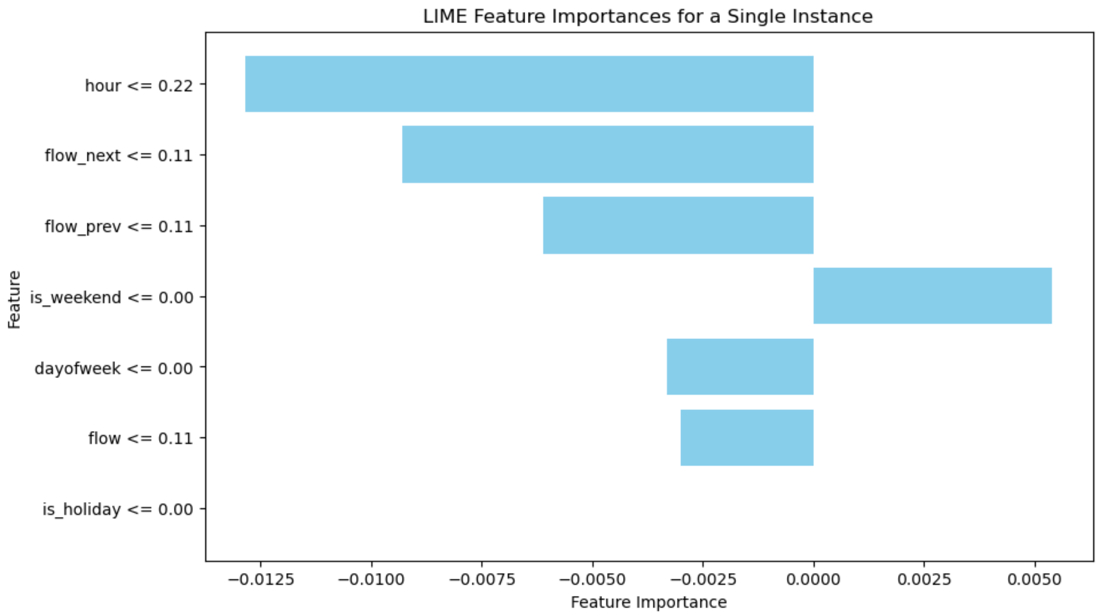



**Figure 9-64: LIME Feature Importance Bar Plot for Athens' case study.**

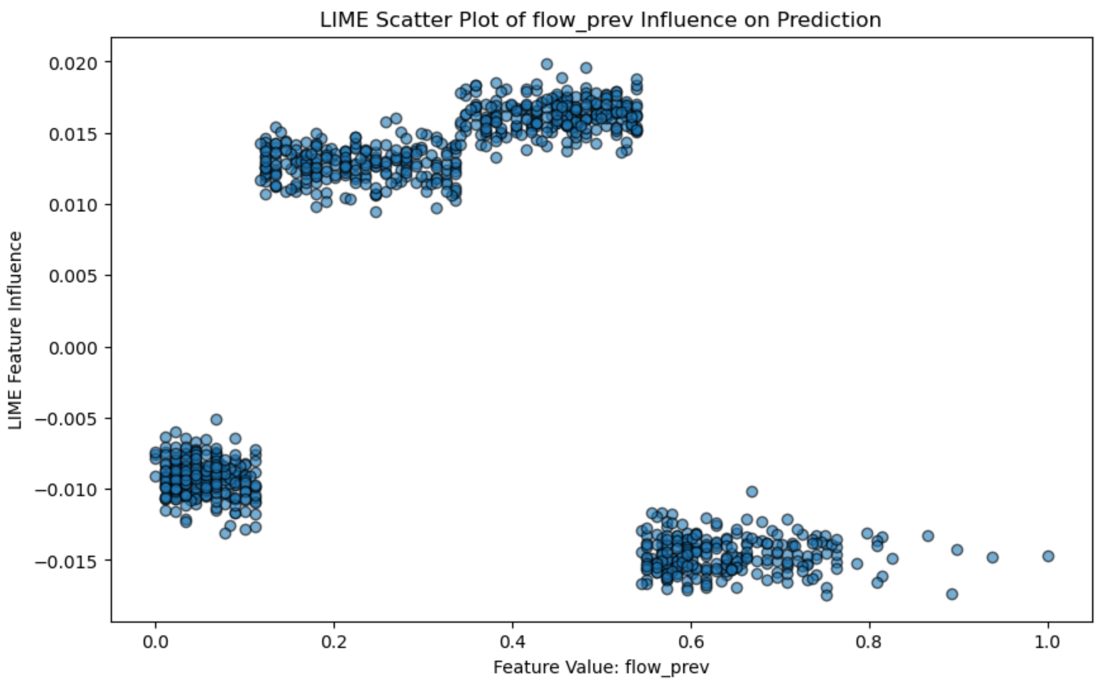



**Figure 9-65: LIME Scatter Plot for specific variable for Athens' case study.**

The results presented in the diagrams presented above reveal key insights into the behaviour of our model and the dataset it was trained on. For instance, the SHAP Waterfall Plot for a particular instance in the dataset clearly indicates how specific features like flow_prev and hour significantly influenced the model's decision, leading to the final prediction. This detailed breakdown allows us to verify that the model's reasoning aligns with domain knowledge and expectations. In the SHAP Summary Plot (Figure 9-61), we observe that certain features consistently exert a strong influence on the model's predictions across the entire dataset. For example, the feature hour emerges as one of the most important predictors, with a wide range of SHAP values indicating its varying impact across different instances. The SHAP Dependence Plot (Figure 9-59) further clarifies how flow_prev interacts with other features, such as dayofweek, influencing the model's predictions in a non-linear manner. This interaction might suggest that the effect of traffic flow on the model’s prediction varies depending on the day of the week, highlighting the importance of capturing such interactions in the model. The SHAP Force Plot (Figure 9-60) concisely summarizes the combined effect of the most influential features for a single prediction, showing how certain features work together to either raise or lower the prediction compared to the base value. This provides a clear and intuitive understanding of the model's decision-making process.

The LIME explanation plot offers a focused view on how the model arrived at a particular prediction by highlighting the top contributing features. For instance, in the analyzed instance, features like hour, flow_prev, and flow_next might emerge as significant contributors to the model’s decision. The LIME plot distinctly shows whether each feature has a positive or negative impact on the prediction and quantifies the magnitude of that impact. This localized explanation allows for a deeper understanding of the model’s behavior in specific cases, which is particularly valuable when decisions need to be explained to non-technical stakeholders or when validating the model against domain knowledge. The LIME diagrams offer various perspectives on how the machine learning model arrived at its predictions, providing both localized and aggregated views of feature importance. The LIME Feature Importance Bar Plot (Figure 9-64) offers a focused view on how the model arrived at a particular prediction by highlighting the top contributing features. For example, in the analyzed instance, features such as hour, flow_prev, and flow_next emerged as significant contributors to the model’s decision. The bar plot distinctly shows whether each feature had a positive or negative impact on the prediction and quantifies the magnitude of that impact. The LIME Explanation Heatmap (Figure 9-63) allows us to compare how features influence predictions across multiple instances. For instance, it may reveal that certain features consistently drive predictions in a particular direction, suggesting a broader pattern in the model's behavior. This comparative analysis is particularly useful for identifying whether the model's decision-making process is consistent and reliable across different cases. The LIME Scatter Plot of Feature Influence (Figure 9-65) provides an insightful view of how changes in a feature's value affect the model's prediction. For instance, the scatter plot may show that as the flow_prev feature increases, its influence on the prediction strengthens in a non-linear manner, indicating a complex relationship between this feature and the model's output.

### 9.6.2 Integration of Human Feedback

Traditional predictive models often struggle in dynamic environments, particularly when dealing with non-recurring incidents—events that are rare or do not follow established patterns. These incidents, by their nature, are challenging to predict because they do not appear frequently enough in historical data for the model to learn from. This is where the integration of human expertise into the machine learning loop, known as Human-in-the-Loop (HITL), becomes invaluable.

Human-in-the-Loop (HITL) is a hybrid approach that combines the strengths of machine learning with human intuition and expertise. In the context of incident management, HITL involves human analysts reviewing and correcting model predictions, especially for cases where the model fails to recognize non-recurring incidents. These human interventions are then fed back into the model during retraining sessions, allowing the model to learn from these corrections and improve its accuracy over time. The HITL approach is particularly beneficial in environments where the data is dynamic, and the nature of incidents can evolve rapidly. By incorporating human feedback into the model's learning process, HITL ensures that the model remains adaptable and continues to improve as new types of incidents emerge.

In the following section, we describe in detail the process of integrating human expertise and feedback in the system, through a series of distinct steps.

1. **System Deployment and Initial Operation**

Our information system is deployed and operational, continuously processing incoming data from various sources and generating real-time incident predictions. The system, initially trained on historical data (from October 2020 to June 2021), monitors and predicts potential incidents as they occur.

2. **Current Operation**

The system is ingesting real-time data from multiple sensors and data sources, using this information to predict non-recurring and recurring incidents. Based on the incoming data, the system generates predictions in real-time, providing crucial insights that inform operational decisions. During this ongoing operational phase, the system occasionally encounters challenges, such as generating false positives (incorrectly predicting incidents) and false negatives (failing to predict actual incidents). These inaccuracies are an expected part of the system's operation and are addressed through a Human-in-the-Loop (HITL) approach.

3. **Human-in-the-Loop (HITL) approach**

To ensure that the system maintains high accuracy, human operators are continuously involved in the prediction process. These experts monitor the system's outputs and intervene whenever discrepancies or inconsistencies are detected.

4. **Real-Time Monitoring**

As predictions are made, human operators review them in real-time. When the system predicts an incident, the operator needs to verify the accuracy of the prediction. If an incident occurs but is not predicted, the operator should flag this as a missed prediction (false negative).

5. **Human Feedback**

Upon identifying false positives or false negatives, operators immediately correct the system's outputs. For instance, if the system incorrectly predicts an incident that does not occur, the operator updates the system with the correct outcome.

Conversely, if the system misses an actual incident, the operator ensures this is logged and corrected in the system's records.

6. **Model Update and Retraining**

The system is designed to adapt continuously by incorporating and storing the corrections and input provided by human operators. This real-time feedback loop enhances the system's predictive capabilities. Moreover, the system undergoes comprehensive retraining periodically using python cron jobs. This process uses the accumulated corrections to refine the model further, ensuring that it adapts to any new patterns or anomalies in the data.

7. **Continuous Evaluation and Improvement**

To maintain the system's effectiveness, the model's performance is continuously evaluated against a stable evaluation dataset (such as data from June 2020). This consistent evaluation ensures that the system not only learns from recent data but also maintains its ability to perform well against established benchmarks.

8. **Real-Time Performance Monitoring**

The system's performance using widely employed metrics, such as precision, recall, and F1-score are monitored, providing feedback on the impact of HITL corrections and retraining. Moreover, the continuous loop of prediction, human correction, and retraining allows the system to adapt to changing conditions quickly, ensuring that its predictions remain reliable and accurate over time.

The detailed methodology for simulating the retraining process within the developed AI-driven traffic incident detection system incorporating human feedback is presented. The objective is to demonstrate how continuous learning, and expert input can enhance the system's accuracy and responsiveness. We utilize existing historical data, deliberately introducing errors from the system's operation, to simulate the real-time update process, for demonstration purposes.

To effectively simulate the learning process, we start by modifying a portion of the historical traffic data to introduce deliberate errors. This simulates an initial state of deployment where any AI system displays certain inaccuracies. As the simulation progresses, we apply updates and human feedback to correct these errors, showcasing the system's capability to learn and improve continuously.

The foundation of any predictive model is the quality and relevance of the data used for training. The first step in the data preparation process involves parsing and aligning the input datasets (operational measurements from loop detectors, network locations and incident records) based on two critical dimensions: time (timestamps) and location (sensor IDs). By aligning the data on these dimensions, we ensure that each record in the dataset represented a unique combination of time and location, capturing the state of the system at each sensor location over time. This alignment is crucial for capturing the spatio-temporal dynamics that are often indicative of incident occurrences.

The baseline incident dataset serves as the ground truth, representing the correct record of incidents. This dataset plays a crucial role in the subsequent phases of model retraining, as it will be used to simulate human intervention, thereby correcting the errors introduced during the operational phases of the predictive system. To effectively evaluate the impact of Human-in-the-Loop (HITL) interventions and retraining, it is essential first to simulate a scenario where the system is prone to errors. This simulation is achieved by artificially introducing systematic errors into a portion of the dataset, specifically targeting the final segment, which reflects the initial deployment phase of the system.

The period selected for the introduction of errors corresponds to the latter part of the dataset, which is indicative of the system's early operational phase. This choice is deliberate, as it allows for a realistic simulation of a newly deployed system that is likely to encounter various predictive inaccuracies.

Those errors are introduced in three primary forms:

- **False Positives:** Incidents incorrectly marked in non-incident data.
- **False Negatives:** Actual incidents that are not marked in the data.

- **Incorrect Incident Details:** Incorrect times and locations of incidents are altered to simulate data inaccuracies.

Using the modified dataset, we train the baseline Machine Learning (ML) and Deep Learning (DL) models which have been thoroughly described as part of Chapter 5. This stage involves:

- **Feature Engineering**: Extracting and engineering key features indicative of traffic incidents, such as traffic flow rates, vehicle speeds etc. To enhance the model's predictive power, additional features are engineered from the raw data. Temporal features are extracted from the timestamp to help the model learn time-based patterns. For instance, certain types of incidents may be more likely to occur during specific hours or on certain days. Additionally, features that captured interactions between different sensors are created, such as differences or correlations between readings from neighboring sensors, providing the model with insights into spatial relationships. More information about the data preprocessing can be found in Chapters 8.1 and 9.2.

- **Model Selection**: Training various machine learning models (e.g. random forests, support vector machines) and DL models (e.g., convolutional neural networks (CNNs) and recurrent neural networks (RNNs)) in addition to the automated ML pipeline as described in Chapter 6, to establish the baseline performance. The baseline performance metrics, such as precision, recall and f1-score, are recorded.

The next phase involves the application of Human-in-the-Loop (HITL) processes to correct the errors introduced into the system. Here, the baseline incident dataset, representing the correct incident records, is utilized to simulate human intervention. For each time segment—specifically on a weekly basis—discrepancies between the error-prone dataset and the baseline dataset are identified. These discrepancies represent the system's misclassifications, which are then systematically corrected by replacing the erroneous labels with the accurate ones from the baseline dataset.

The framework continuously updates the models based on new data and simulated human feedback. Key components of this process include:

1. **Batch Updates:** Periodically, the system performs batch updates where a larger set of accumulated feedback data is used to retrain the model. This helps to consolidate learning and reinforce correct patterns. This retraining simulates the system's iterative learning process, where human feedback is continuously integrated to refine and improve its predictive capabilities. This process is repeated for each subsequent defined time period, allowing for a progressive enhancement of the model's accuracy.

2. **Model Validation:** After each update, the models are validated using a validation set to ensure that the updates have improved performance. Moreover, the established key performance metrics are monitored continuously.

To determine the optimal retraining strategy, the impact of different retraining frequencies is assessed:

1. **Weekly Retraining**: The weekly retraining strategy serves as the primary approach, balancing the need for regular updates with computational efficiency. This strategy allows for the accumulation of sufficient data corrections over a week, potentially leading to more stable improvements in model performance.

2. **Biweekly Retraining**: By contrast, biweekly retraining tests the model's ability to adapt when corrections are less frequent. This approach may offer advantages in terms of computational resource management, but it may also slow the model's adaptation to new patterns.

Continuous monitoring of performance metrics is essential throughout the simulation. By evaluating these metrics at regular intervals, we can assess improvements and compare them against the initial baseline performance. This step highlights the effectiveness of retraining and human feedback in enhancing the system.

#### 9.6.2.1 Example scenario

To illustrate the methodological process, we present an example scenario.

***Initial Baseline Training***

The initial model was trained using data stemming from case study I (Athens) from October 2020 to June 2021. This period provided a substantial amount of historical data, allowing the model to learn typical incident patterns in a controlled environment, free from the errors associated with early-stage deployment. Specifically, the algorithm selected for these experiments is the RandomForest, since it has been shown in other experiments (ref to Chapter 9.4.1) to perform extremely well against the benchmarks established.

To establish a robust baseline, the model was evaluated using data from September 2021. This dataset serves as a consistent reference point for all subsequent evaluations. The September 2021 dataset is used to establish the model's baseline performance. It is then used repeatedly to evaluate the model’s performance after errors are introduced and after each retraining session. Key performance metrics, including precision, recall, and F1-score, were recorded. For more information regarding these metrics, the reader is invited to refer to Chapter 8.1.1.7.1. These metrics form the foundation against which all future comparisons will be made, ensuring that any improvements or deteriorations in model performance can be traced back to the effects of HITL corrections and retraining.

***Introduction of System Errors***

The system was simulated to go live in July 2021, at which point it began making errors typical of an early-stage predictive model. To replicate this scenario, the predictions produced by the system (with the respective errors) were introduced into the July-August 2021 dataset. Specifically, these errors correspond to the system's potential to either overestimate or underestimate the likelihood of incidents during its phase of operation, in the initial stages. The resulting metrics highlighted the extent to which the model's predictions and the errors produced were compromised by the inaccuracies.

***Human Feedback Integration***

Following the system's errors in the July-August 2021 dataset, a HITL approach was employed. Errors were corrected in a simulated manner using the baseline incident data as the ground truth. Thus, the operator feedback was simulated with 100% accuracy for approving/rejecting incidents, while missed incidents are also included. This correction process simulated the real-world scenario where human operators would intervene to rectify the system's mispredictions.

***Retraining and Model Updates***

Models are retrained weekly (also biweekly) using accumulated feedback. After correcting the errors for each week in the July-August 2021 dataset, the model was retrained periodically. This retraining process aimed to incorporate the newly corrected data, allowing the model to adapt and improve its predictive accuracy. Importantly, each retraining cycle was followed by an evaluation and comparison against the consistent September 2021 dataset.

These human-corrected labels were then used in the model's weekly retraining process. By incorporating this feedback, the model could learn from its mistakes and improve its ability to predict non-recurring incidents in future iterations. The retraining process involved updating the model with new correct data each week, simulating the human corrections, ensuring that the model continuously adapted to new patterns in the data.

### *Performance Monitoring*

By maintaining September 2021 as the evaluation set throughout the retraining process, we ensured that performance improvements could be tracked reliably. After each retraining iteration, performance metrics were recorded and compared against the baseline established earlier. This method allowed for a clear assessment of how the model's ability to predict incidents evolved as it learned from the HITL corrections.

### *Assessing Final Model Performance*

Upon completing the retraining sessions, the final model was once again evaluated on the June 2021 dataset. This final evaluation provided a direct comparison between the model's initial baseline performance, and its improved state after HITL correction and retraining.

This approach demonstrates the cumulative effect of weekly retraining on restoring the model's performance to its original state, as measured by a consistent evaluation set.

## Evaluation and Comparison of Retraining Frequencies

#### *Weekly Retraining*

The standard retraining approach involved updating the model on a weekly basis. This frequency was chosen to balance the need for model adaptation with the stability required for accurate predictions. Weekly retraining allowed the model to incorporate sufficient new data and human corrections, leading to noticeable improvements in performance, particularly in the recall score for non-recurring incidents. In Table 9-12, the results from the experiments are illustrated. Moreover, in Figure 9-66 the process of the retraining, in addition to the datasets used for training and test are shown.

Table 9-12: Results of performance metrics during retraining for 8 weeks.

| Evaluation Stage | Precision | Recall | F1-Score |
|---|---|---|---|
| **Initial Predictions** | 0.65 | 0.55 | 0.60 |
| **HITL Retraining After Week 1** | 0.67 | 0.58 | 0.62 |
| **HTIL Retraining After Week 2** | 0.71 | 0.61 | 0.66 |
| **HITL Retraining After Week 3** | 0.72 | 0.61 | 0.66 |
| **HITL Retraining After Week 4** | 0.74 | 0.64 | 0.69 |
| **HITL Retraining After Week 5** | 0.74 | 0.68 | 0.71 |
| **HTIL Retraining After Week 6** | 0.76 | 0.68 | 0.72 |
| **HITL Retraining After Week 7** | 0.76 | 0.70 | 0.73 |
| **HITL Retraining After Week 8** | 0.78 | 0.72 | 0.75 |

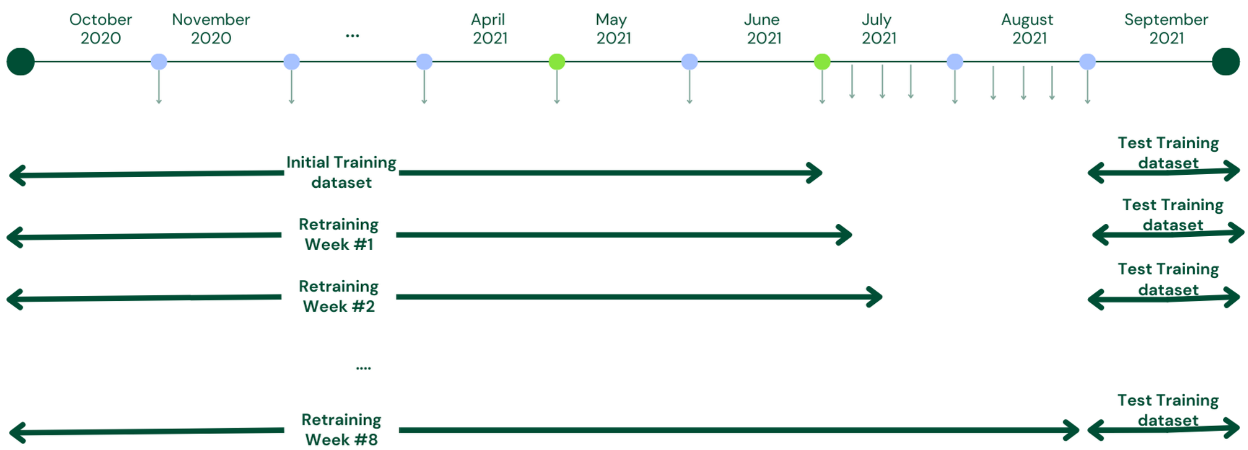



Figure 9-66: Illustration of datasets in the retraining process.

***Biweekly Retraining***

To further explore the impact of retraining frequency on model performance, additional experiments were conducted with biweekly retraining schedules. Daily retraining would have provided the model with the most up-to-date data and corrections, potentially improving responsiveness to new incident patterns. However, this approach also risked overfitting to short-term trends and required significant computational resources.

Biweekly retraining offers a more conservative approach, focusing on broader trends rather than daily fluctuations, as shown in Table 9-13. While this approach reduces computational overhead, it shows to have limited the model’s ability to quickly adapt to new patterns, particularly in environments where incident characteristics could change rapidly.

**Table 9-13: Results of performance metrics for biweekly retraining in the period of 8 weeks.**

| Evaluation Stage | Precision | Recall | F1-Score |
|---|---|---|---|
| **Initial Predictions** | 0.65 | 0.55 | 0.60 |
| **HITL Retraining After Biweekly Period 1** | 0.71 | 0.61 | 0.66 |
| **HTIL Retraining After Biweekly Period 2** | 0.74 | 0.64 | 0.69 |
| **HITL Retraining After Biweekly Period 3** | 0.76 | 0.68 | 0.72 |
| **HITL Retraining After Biweekly Period 4** | 0.78 | 0.72 | 0.75 |

***Comparative Analysis***

The use of the validation dataset as a stable reference point allowed for precise comparisons across different stages of the model’s development. The findings underscore the value of incorporating human expertise into the machine learning loop, particularly in scenarios where early-stage systems are prone to errors. This approach not only facilitated the identification and correction of predictive inaccuracies but also provided a framework for continuous improvement through systematic retraining.

The comparative analysis of these retraining frequencies revealed that weekly retraining provided the optimal balance between responsiveness and stability. The model was able to adapt to new patterns effectively without the risk of overfitting or excessive computational demands.

Moreover, several constatations have been made. The model's precision, recall, and F1-score exhibit a marked improvement following each retraining cycle. This improvement is particularly anticipated in the recall metric, which measures the model's ability to correctly identify actual incidents, including those previously misclassified. Moreover, by systematically comparing the results of weekly, and biweekly retraining, we have identified the most effective retraining strategy. We have not chosen daily retraining, since it is believed that it may lead to rapid performance gains but could also risk overfitting or require significant computational resources. Weekly retraining provides a balanced approach, offering consistent improvements without excessive computational demands. Biweekly retraining, while potentially more resource-efficient, has been shown to delay the model's ability to incorporate new patterns promptly.

## 9.7 Discussion of results and Comparison between urban environments

In general, evaluation results need to be understood from the perspective of the nature of the data and methodologies used. We must remember that labelled incidents were limited to visible areas of the network, therefore the false positive rate is seriously affected. There is no certainty that false positives might be due to an invisible event to the data supplier (therefore, there was an event, but it was not labelled) or truly a faulty prediction by the algorithm.

### 9.7.1 Discussion of Results for Unplanned Incident Detection

For the detection of unplanned incidents, the evaluation dataset has been formatted in 5 minutes intervals to feed the classification algorithms, therefore when computing precision there is a true positive when the event is detected at exactly the annotated timestamp. However, in real scenarios an ideal incident detection algorithm should be able to spot an event ideally before it happens or at least within a reasonable time margin to allow for proactive intervention.

#### 9.7.1.1 Performance of Traditional ML and DL Models

Regarding the results obtained from the experiments in both case studies, we deduce that there is some fluctuation on the results, and that some of the algorithms show low performance on the evaluation metrics. The best performing algorithm is the Support Vector Machine both for the Athens and the Antwerp datasets, having achieved both high precision and recall. This is in line with the literature which supports that SVM generally perform very well when the labelled incidents dataset is available, since they work in a supervised manner. On the other hand, this approach depends on the existence and reliability of an incident dataset, and it is possible that the model would overfit the dataset and have trouble when encountering new unseen samples.

Moreover, BCNN and WNN manage to reach a very high recall but are unable to perform well in terms of precision (and thus the F1-score is also impacted). However, we are able to discern that the wavelet transformation performs slightly better than the BCNN, also confirming the findings in literature which make use of the wavelet transformations for time series datasets. Concerning the Autoencoder, the results as shown above could definitely be improved. The results obtained could probably be due to the fact that the architecture of the autoencoder we have employed is quite simple. One of the further improvements could include the addition of new layers in the architecture and the additional fine-tuning of these layers in order to achieve better results.

The Isolation Forest algorithm is able to capture a satisfactory recall, but the precision achieved is very low (thus impacting the F1-score obtained). The Bidirectional LSTM manages to deliver one of the best results in terms of precision and quite good results in terms of recall. We have experimented with various architectures for the LSTM, and finally the best performance was obtained by employing a deep stacked bidirectional and unidirectional LSTM neural network architecture. This considers both forward and backward dependencies in time series data and predicts traffic flow. Finally, the model is able to capture the points where the predicted and actual values were significantly different, and a threshold for loss value was set based on the history of loss values in training and testing to capture the incidents.

Aimsun's algorithm shows low performance on the established metrics, however when analyzing results on an event-based rationale for the Athens use case (within a time margin of 15 minutes around the labelled event), Aimsun's system was able to detect 11 events out of a total of 15 in the analyzed period (May 2021) yielding a recall of 73% which is an acceptable level of performance for non-recurrent events. However, one of Aimsun baseline's limitation is the fact that it is bound to produce a high number of false positives as shown by precision results of 8% (17% in an event-based evaluation).

#### 9.7.1.2 AutoML Performance and Comparison with Traditional Models

The application of AutoML across Athens and Antwerp demonstrated its adaptability but also highlighted its limitations. In Athens, where congestion patterns were relatively stable, AutoML-tuned models achieved high precision and recall, demonstrating strong generalization to traffic conditions. However, in Antwerp, AutoML models required additional fine-tuning to maintain their predictive performance. These findings suggest that while AutoML provides a robust framework for incident detection, localized calibration is necessary to account for environmental variability.

Despite its advantages, AutoML presented computational trade-offs that should be considered for real-time applications. The increased computational cost and training time were particularly evident when optimizing deep learning models such as LSTM, where AutoML required longer processing times compared to traditional ML models. While the added complexity led to improved forecasting performance, real-world deployment may require a balance between accuracy and computational feasibility, especially when models need frequent retraining.

In this light, among the algorithms tested, for the Athens dataset, we observe significant performance superiority of the AutoML approach employed compared to the baseline methods. For the Antwerp dataset, among the algorithms tested, the Support Vector Machine (SVM) emerged as the top-performing method, demonstrating high precision and recall, while the Autoencoder demonstrated high recall rates but struggled to achieve satisfactory precision, indicating a tendency to flag non-incident anomalies as potential traffic disruptions.

Lastly, it is important to recognize that while AutoML aims to streamline and optimize the model selection and training process, it does not negate the value of understanding specific ML techniques' performance in targeted applications. Our comparison seeks to highlight how our AutoML-powered approach stand against manually tuned and selected models in the specific domain of incident detection using loop detector data, emphasizing the efficacy, adaptability, and performance in real-world scenarios. Among the tested models, AutoML consistently selected ensembles and tree-based algorithms such as Random Forest and Gradient Boosting, while it also prioritized LSTM architectures for handling sequential traffic data, demonstrating AutoML's capability to identify well-suited architectures for different types of incident detection tasks. To summarize, The Athens dataset showed clear advantages of AutoML over baseline models, while in Antwerp, SVM achieved the highest F1-score and recall, but AutoML outperformed SVM in precision. These variations highlight the necessity for algorithmic adaptation based on local traffic conditions and data availability across different urban environments.

#### 9.7.1.3 Final considerations and limitations

All in all, one of the limitations of our analysis in both cases is the fact that these techniques are bound to produce a high number of false positives as shown by the precision results, thus the concept of false positives is hard to really assess in the incident detection task due to possible blind spots in the network. It is evident that each city presented unique challenges and outcomes in incident detection for unplanned incidents.

However, we acknowledge that our evaluation primarily focuses on conventional metrics such as precision, recall, and F1-score. While these metrics provide valuable insights into the models' performance in detecting incidents accurately, they do not fully encapsulate the operational context within which these detection models are deployed. Factors such as the mean time to detect an incident, the speed of propagation of detected incidents, and the practical implications of false positives and false negatives on traffic management and response strategies are not directly addressed as part of this PhD dissertation.

### *9.7.2 Discussion of Results for Planned Incident Detection (Recurring Congestion case)*

The identification and forecasting of planned incidents, particularly in the context of recurring congestion, were explored through a combination of time-series forecasting models, deep learning techniques, clustering analyses, and visual analytics. The findings across different case studies confirmed that congestion follows predictable temporal and spatial patterns, making it possible to anticipate its occurrence with high accuracy. By analyzing historical traffic flow, occupancy, and speed data, the study demonstrated that congestion patterns could be effectively modeled, allowing for early detection and intervention.

A major part of the analysis focused on time-series forecasting models, where Autoregressive Integrated Moving Average (ARIMA) and Long Short-Term Memory (LSTM) networks were employed. ARIMA provided a strong baseline in cases where traffic exhibited stable seasonal variations, effectively capturing long-term congestion cycles. However, its reliance on linear assumptions made it less effective in scenarios with highly dynamic congestion patterns. In contrast, LSTM networks, with their ability to capture long-term dependencies in traffic data, outperformed ARIMA in most cases, particularly in urban environments characterized by high variability in traffic flow. The results demonstrated that LSTM-based predictions aligned closely with actual congestion trends, making them a robust approach for modeling planned traffic conditions.

To enhance spatial insights into congestion dynamics, clustering methods were employed to categorize congestion patterns across different road segments.

Heatmap visualizations and box plots revealed consistent congestion-prone areas, reinforcing the hypothesis that specific locations experience recurring congestion cycles. These findings were further supported by correlation analyses, which confirmed strong relationships between traffic indicators such as speed, flow, and occupancy. The ability to visually interpret these relationships strengthened the validity of the forecasting models, as they provided clear evidence of how congestion evolved in different urban settings.

The evaluation of these methodologies in real-world case studies further validated their effectiveness in predicting planned incidents. The results showed that applying advanced analytics-driven models enabled accurate anticipation of congestion before it became disruptive, demonstrating the potential for proactive traffic management strategies. The deep learning-based approaches, particularly LSTM networks, consistently performed better than traditional statistical methods, offering improved predictive accuracy in cases where traffic conditions changed dynamically. However, the dependency on high-quality input data was evident, as data sparsity and sensor inconsistencies presented challenges in some instances. Missing values affected the generalization capability of forecasting models, highlighting the need for careful data preprocessing and sensor coverage optimization.

The comparative analysis between the two case studies in Athens and Antwerp revealed that while congestion trends followed distinct characteristics in each city, their recurring nature allowed for effective modeling. In Athens, congestion was largely influenced by urban mobility patterns, with peak-hour traffic following well-defined cycles. In Antwerp, where major roadways and port activity contributed to congestion, the predictive models adapted to different traffic flow behaviors.

Overall, our work confirmed that a multi-method approach, combining time-series forecasting, clustering, and visual analytics, provides a comprehensive understanding of recurring congestion patterns and planned incident detection. The extracted results emphasized the importance of model adaptability, high-resolution traffic data, and spatial-temporal consistency in ensuring accurate and scalable congestion forecasting. While deep learning models proved to be highly effective, their success depended on the availability of reliable input data and the ability to fine-tune models to specific urban environments.

### *9.7.3 Discussion of Results for the Integration of HITL and Explainability*

The inclusion of explainability features as part of the system helps ensure the trust of the system by its users. Moreover, the integration of a Human-in-the-Loop methodology and regular retraining significantly enhanced the model's ability to predict non-recurring incidents. The HITL approach, in particular, played a crucial role in improving the model's f1 score, enabling it to better identify rare events that traditional models might overlook.

The comparison of different retraining frequencies provided valuable insights into the optimal approach for maintaining high prediction accuracy while balancing the need for computational efficiency. The findings suggest that weekly retraining strikes the best balance, allowing the model to stay up to date with new patterns without overfitting or resource exhaustion. Our approach demonstrates the effectiveness of combining machine learning with human expertise in dynamic environments, particularly when dealing with non-recurring incidents. By continuously incorporating human feedback and adjusting the retraining frequency to suit the specific needs of the environment, predictive models can remain accurate and reliable, even in the face of evolving data patterns.

While our approach demonstrates the effectiveness of combining machine learning with human expertise in dynamic environments, several avenues for future research can be explored. First, the integration of more advanced explainability techniques (saliency maps, counterfactual explanations etc.) could provide deeper insights into how models make predictions, further enhancing trust and usability for human operators. These explainability features can help identify biases or gaps in model performance, facilitating targeted improvements over time.

Another promising direction is the exploration of adaptive retraining schedules. Rather than adhering to a fixed retraining frequency (e.g., weekly), models could benefit from adaptive retraining that triggers updates based on the detection of significant data shifts or anomalies. This would allow the system to stay responsive to changes without unnecessary resource expenditure.

Moreover, an approach which could be used is the incorporation of online learning methodologies, techniques where machine learning models continuously update their parameters as new data becomes available, rather than relying on traditional batch learning where models are retrained from scratch periodically. This is particularly useful in dynamic environments, such as transport systems, where conditions change frequently due to new traffic patterns, incidents, or weather conditions. Online learning enables models to adapt to changes in real-time, improving their responsiveness and accuracy. In the context of incident detection systems, online learning allows the model to adjust its predictions as it receives new traffic data, making the system more resilient to evolving patterns. Furthermore, online learning methodologies reduce computational costs because only new data is used for updating the model, rather than requiring full retraining, which can be resource-intensive.

Unlearning methodologies, on the other hand, are techniques that allow models to "forget" specific parts of learned data when they are no longer relevant or if they are incorrect. This is particularly important for privacy concerns, where models might need to forget sensitive data (for instance during crises, such as COVID-19), or in situations where new information renders old patterns invalid, such as changes in road infrastructure. In the context of transportation systems, unlearning could be used to ensure that models no longer rely on outdated traffic patterns that no longer apply, ensuring the predictions remain relevant and accurate.

Finally, future research could explore automated feedback loops that gradually reduce the need for human intervention by learning from historical feedback. Over time, these systems could become increasingly autonomous while maintaining a high level of accuracy and explainability, thus improving the overall efficiency and reliability of incident detection in real-world applications.

# 10 Conclusions and Future Work

## 10.1 Conclusions

The work presented in this PhD dissertation marks significant progress in the conceptual engineering, development and deployment of a comprehensive AI-driven incident detection system for urban traffic management. Through the integration of advanced AI techniques, traditional traffic management methods, automated Machine Learning, Explainability and Human-In-The-Loop (HITL) concepts, the research has demonstrated the potential for creating a robust and adaptable solution for real-time traffic incident detection. The key conclusions which emerge from our work done can be summarized in the following:

- **Introduction and Motivation (Chapter 1):** This chapter sets the foundation for the research by discussing the current challenges and issues in traffic management systems, addressing their limitations, especially in urban environments where complex and dynamic traffic patterns challenge traditional systems. The motivation of the work conducted is rooted in the need for automated yet adaptable solutions that can integrate human insights, leading to a new AI-based, human-centric approach for real-time traffic incident detection.
- **Literature Review (Chapter 2):** An extensive review of existing traffic incident detection methods and technologies is provided here. Key algorithms, from comparative methods to advanced machine learning models, are examined, highlighting their strengths and weaknesses. The literature review also points out the existing research gaps, which this dissertation has aimed to address through novel methodologies.
- **Research Challenges (Chapter 3):** This chapter outlines the specific research questions driving the study. These questions are focused on developing a comprehensive framework that combines automated and human-centered

approaches for real-time traffic monitoring. The goal is to enhance incident detection for both planned and unplanned events, by using state-of-the-art methods and algorithms, improve transparency through explainability techniques, involve human operator expertise and achieve high performance across different urban environments.

- **Framework for Real-Time Monitoring and Prediction of Traffic Incidents (Chapter 4):** Introducing a multi-component framework, this chapter presents the four key pillars where our work is based upon: Data Analytics, Automated Machine Learning (AutoML), Human-in-the-Loop (HITL), and Explainability. Each pillar plays a crucial role in building a robust and adaptive system, with AutoML ensuring automation in model development, HITL enhancing adaptability and performance, and explainability fostering trust and transparency.
- **AI-Driven Traffic Incident Detection for Planned and Unplanned Events (Chapter 5):** Focusing on data-driven methods, this chapter delves into advanced analytics techniques for detecting unplanned incidents and identifying recurring congestion patterns. It showcases the effectiveness of machine learning and deep learning models in handling diverse types of incidents, enhancing the adaptability and accuracy of traffic incident detection.
- **AutoML-Driven Incident Detection (Chapter 6):** This chapter presents the use of TPOT within AutoML for automating model development and optimization. By automating feature selection, model tuning, and pipeline creation, TPOT reduces the need for manual adjustments, thereby streamlining the model development process.
- **Human-in-the-Loop and Explainability in Incident Detection (Chapter 7):** Highlighting the importance of HITL, this chapter emphasizes the integration of human feedback in refining AI models. Explainability techniques, such as SHAP and LIME, are implemented to ensure that model predictions are understandable to operators. This approach not only enhances transparency but also empowers operators to make informed decisions during critical times.

- **Information System AutoEventX (Chapter 8):** Detailing the technical architecture of AutoEventX, this chapter outlines how the system combines data pre-processing, model development and training, model evaluation and real-time analytics in a scalable, operator-friendly platform. AutoEventX's functionalities are demonstrated through real-world cases, which illustrate its effectiveness in incident detection and its flexibility in urban traffic management scenarios.
- **Deployment and Evaluation in Real-world Case Studies (Chapter 9):** This chapter presents an in-depth evaluation of AutoEventX, tested in the cities of Athens and Antwerp. The results confirm the system's adaptability to different traffic conditions, its robust performance across varied environments, and the value of including automated and HITL approaches in improving detection accuracy.
- **Conclusions and Future Work (Chapter 10)**: Summarizing the dissertation's contributions, this chapter discusses also the limitations of the current study and proposes future directions. Future research should focus on integrating multi-source data, advancing human-AI collaboration, enhancing explainability, and incorporating prescriptive analytics to optimize automatic traffic incident detection in smart mobility systems.

## *10.2 Limitations*

Generally, data-driven methods used in the context of automatic incident detection algorithms (AIDA) have their limitations. Perhaps the most commonly cited limitation is that many IDAs are unable to differentiate incidents from contexts, resulting in a high false alert rate. Noise from signals on junctions can cause congestion similar to that of an incident, leading to false alerts in traffic variable based AIDAs. Finally, many AIDAs are only capable of indicating when an incident has taken place in the vicinity of a detector. Traffic operators could respond more effectively if the exact incident location and expected congestion propagation could be estimated.

These features are closely related to incident detection and could be accounted for within the design of IDAs to aid operators further.

Apart from these well-known limitations, it is noteworthy to mention some limitations specific to our work. Limitations arise from the manual registration of incidents by operators, potentially leading to omissions or timing inaccuracies in the dataset. Consequently, such discrepancies can significantly skew the performance evaluation of our algorithms. The filtering process specifically for the Athens labelled incidents' dataset is susceptible to errors, and despite being informed by stakeholder operator expertise, the risk of overlooking significant incidents cannot be ignored. Additionally, the decision to format the evaluation dataset in 5-minute intervals may seriously have affected the precision metric, as it requires the detection of events at their exact recorded timestamps.

Moreover, our analysis is contingent upon the data quality and reliability, and more specifically on the measurements of the detectors, and the accuracy and completeness of incident reporting, the basis of our evaluation. False positives within our model outcomes may not solely represent algorithmic inaccuracies but could also reflect events unlabeled due to visibility issues within the network's coverage. The reliance on a single data source, CCTV footage, limits our ability to comprehensively capture all incidents, suggesting that incorporating diverse data types could enhance detection and reduce false positives. The challenge of accurately determining false positives due to potential network blind spots is recognized and underscores the need for a multifaceted approach in future research to mitigate such issues. One way to decrease the false alerts generated is to incorporate various types of data, such as CCTV cameras which is also the vision of our work for the future, to train the algorithms. Despite the abundance of available data and the advanced capabilities of machine learning algorithms, only a limited number of studies have effectively utilized the combination of multiple data sources, as stated by the review conducted by Kashinath et al. (Kashinath, et al., 2021).

Lastly, we recognize that our work's scope is limited by its focus on immediate detection metrics (precision, recall and f1-score), which, while crucial, do not encompass the broader operational implications of deploying such technologies in complex traffic management systems. As such, we propose that future work should extend beyond traditional performance metrics to evaluate AutoML and other ML techniques within the context of their downstream applications. Specifically, research should explore the operational impact of these detection technologies, including their effect on mean time to detect incidents, the propagation speed of detected incidents through traffic networks, and their integration into comprehensive traffic management strategies. Such an approach will provide a more holistic understanding of the value and limitations of AutoML and other ML techniques in traffic incident detection, guiding both technological development and strategic implementation in this critical domain.

## *10.3 Future work*

While significant progress has been made in integrating machine learning, data explainability, and human feedback into incident detection processes, several areas remain open for further exploration and refinement. Future research directions in this domain can further enhance the accuracy, reliability, and trustworthiness of our own system but also traffic incident detection systems in general. Below we present some key areas where future research can make substantial contributions:

1. **Multi-source data integration**: Incident detection can be significantly improved by incorporating diverse data sources such as CCTV feeds, crowdsourced reports, weather data, social media feeds and real-time sensor data. Future studies should focus on developing improved data fusion methodologies and algorithms to enhance accuracy and robustness.
2. **Inclusion of more advanced algorithms and tools**: Future work should explore reinforcement learning techniques and domain adaptation methods that can

enhance the efficiency and adaptability of AI models specifically for the task at hand.

3. **Operational impact of automatic incident detection technologies:** Future work should explore how the different technologies used in the context of our framework influence the mean time to detect incidents, the propagation speed of detected incidents through traffic networks, and their integration into comprehensive traffic management strategies. Understanding these aspects will help in assessing the real-world effectiveness of AI-driven detection systems.

4. **Advanced Human-AI Collaboration and HITL features:** Future research can explore more sophisticated methods of human-AI collaboration, where the interaction between human operators and AI systems is more seamless and intuitive. This includes developing user potentially integrating augmented reality (AR) or virtual reality (VR) to provide operators with more interactive tools for incident management. (Olugbade, Ojo, Imoize, Isabona, & Alaba, 2022) (ElSahly & Abdelfatah, 2022) Also, developing real-time feedback mechanisms would allow operators to iteratively improve model accuracy. Implementing adaptive learning algorithms that can continuously learn from human feedback and real-world data is a crucial area for future exploration. This involves developing models that can dynamically update their parameters and improve their performance over time, based on the feedback received from operators and the outcomes of previous predictions. (ElSahly & Abdelfatah, 2022) (Olugbade, Ojo, Imoize, Isabona, & Alaba, 2022). Extending the AutoEventX pipeline to include techniques such as unlearning, adaptive retraining schedules, and semi-supervised feedback mechanisms will enable models to evolve dynamically while minimizing the annotation burden on human operators.

5. **Enhanced Explainability Techniques:** While LIME and SHAP are currently used for providing explanations, there is room for improvement in making these explanations more comprehensive and accessible to non-expert users. Future

research can focus on developing new explainability techniques that offer deeper insights into model decisions, are easier for operators to understand, and can be customized to different user needs (ElSahly & Abdelfatah, 2022) (Olugbade, Ojo, Imoize, Isabona, & Alaba, 2022).

6. **Introduction of prescriptive analytics:** Moving beyond predictive analytics, prescriptive analytics can play a crucial role in incident management by suggesting actionable interventions to mitigate congestion and reduce the overall impact of incidents. Future research should investigate how AI can recommend and implement optimal traffic control measures in real time.

7. **Integration with Other Smart City Systems:** Expanding the integration of AI-driven traffic incident detection systems with other smart city infrastructures can enhance the overall efficiency of urban management. This includes connecting traffic management systems with public transportation, emergency response, and environmental monitoring systems to create a more holistic approach to urban incident management. (Liang, et al., 2022)

8. **Ethical and Privacy Considerations:** As AI systems become more integrated into traffic management, it is essential to address ethical and privacy concerns. Future research should focus on developing frameworks that ensure data privacy and address potential biases in AI models. This includes creating transparent policies for data usage and developing algorithms that are fair and equitable. (Olugbade, Ojo, Imoize, Isabona, & Alaba, 2022)

In summary, this research has introduced a comprehensive framework for leveraging AI-driven methodologies in traffic incident detection, incorporating explainability, human-in-the-loop processes, and prescriptive analytics. While the proposed solutions offer significant improvements, ongoing advancements in data integration, model adaptability, and explainability will be necessary to fully realize the potential of AI in smart transportation systems. By addressing these challenges, future research can further refine and enhance the effectiveness of incident detection systems, ensuring their integration into sustainable and intelligent urban mobility solutions.

# References


Hireche, S., & Dennai, A. (2020). A Review of Artificial Intelligence Techniques used for Urban Automatic Incident Detection Systems. *ICSCA 2020: Proceedings of the 2020 9th International Conference on Software and Computer Applications* (pp. 281–286). New York, NY, USA: Association for Computing Machinery.

Li, L., Lin, Y., Du, B., Yang, F., & Ran, B. (2022). Real-time traffic incident detection based on a hybrid deep learning model,. *Transportmetrica A: Transport Science, 18:1, 78-98*.

Evans, J. R. (2020). *Improving road incident detection algorithm performance with contextual data.*

Hireche, S., & Dennai, A. (2020). Machine Learning Techniques for Road Traffic Automatic Incident Detection Systems: A Review. *Springer Nature Switzerland AG 2020 M. Hatti (Ed.): ICAIRES 2019, LNNS 102, pp. 60–69.*

Balke, K. N. (1993). An evaluation of existing incident detection algorithms. *TRID*.

Khoury, J. A., Haas, C. T., Mahmassani, H., & Logman, H. (2003). Performance Comparison of Automatic Vehicle Identification and Inductive Loop Traffic Detectors for Incident Detection. *Journal of Transportation Engineering / Volume 129 Issue 6*.

Dudek, C. L., Messer, C. J., & Nuckles, N. B. (1974). Incident detection on urban freeways. *Transportation Research Record 495, 12–24*.

Liu, G., Jin, H., Li, J., Hu, X., & Lia, J. (2022). A Bayesian deep learning method for freeway incident detection with uncertainty quantification. *Accident Analysis & Prevention, 176*.

Zhu, L., Guo, F., Krishnan, R., & Polak., J. W. (2018). *The Use of Convolutional Neural Networks for Traffic Incident Detection at a Network Level., Technical Report.*

Li, D., Hu, X., J. C., & Zhou, J. (2017). *Learning to detect traffic incidents from data based on tree augmented naive bayesian classifiers. Discrete Dyn. Nat. Soc. 2017, 1–9 .*

Dardor, M., Chlyah, M., & Boumhidi, J. (2018). *Incident detection in signalized urban roads based on genetic algorithm and support vector machine. In: Proceedings of the 2018 International Conference on Intelligent Systems and Computer Vision, Fez, Morocco, 2–4 .* IEEE.

Shang, Q., Feng, L., & Gao, S. (2020). A Hybrid Method for Traffic Incident Detection Using Random Forest-Recursive Feature Elimination and Long Short-Term Memory Network With Bayesian Optimization Algorithm. *IEEE Access PP(99).*

Zhu, L., Wang, B., Yan, Y., Guo, S., & Tian, G. (2022). A novel traffic accident detection method with comprehensive traffic flow features extraction. *Signal, Image and Video Processing*.

Li, D., Chen, D., Jin, B., Shi, L., Goh, J., & Ng., S.-K. (2019). MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks. *International Conference on Artificial Neural Networks.* (pp. 703-716). Springer.

Cui, Z., Ke, R., & Wang, Y. (2018). Deep Stacked Bidirectional and Unidirectional LSTM Recurrent Neural Network for Network-wide Traffic Speed Prediction. *6th International Workshop on Urban Computing (UrbComp 2017).*

Zhou, Z., Wang, Y., Xie, X., Chen, L., & Liu, H. (2020). RiskOracle: A minute-level citywide traffic accident forecasting framework. *Proceedings of the AAAI conference on artificial intelligence, 34*(01), 1258-1265.

Yu, L., Du, B., Hu, X., Sun, L., Han, L., & Lv, W. (2021). Deep spatio-temporal graph convolutional network for traffic accident prediction. *Neurocomputing, 423*, 135-147.

Wang, B., Lin, Y., Guo, S., & Wan, H. (2021). GSNet: learning spatial-temporal correlations from geographical and semantic aspects for traffic accident risk forecasting. *Proceedings of the AAAI conference on artificial intelligence, 35*(5), 4402-4409.

Hall, F. L., Shi, Y., & Atala, G. (1993). On-line Testing of the McMaster Incident Detection Algorithm under Recurrent Congestion.

Samant, A., & Adeli, H. (2000). Feature Extraction for Traffic Incident Detection Using Wavelet Transform and Linear Discriminant Analysis. *Computer-Aided Civil and Infrastructure Engineering 15 (4): 241–250.*

Thancanamootoo, S., & Bell, M. (1988). Automatic detection of traffic incidents on a signalcontrolled road network. *Transport Operations Research Group report 76(76).*

Sheu, J.-B., & Ritchie, S. G. (1998). A new methodology for incident detection and characterization on surface streets. *Transportation Research Part C: Emerging Technologies 6(5), 315– 335.*

Lee, J.-T., & Taylor, W. C. (1999). Application of a dynamic model for arterial street incident detection. *ITS Journal - Intelligent Transportation Systems Journal 5(1), 53–70.*

Yuan, F., & Cheu., R. L. (2003). Incident Detection Using Support Vector Machines." Transportation Research Part C: Emerging Technologies 11 (3–4): 309–328.

Xiao, J., & Liu., Y. (2012). Traffic Incident Detection Using Multiple-kernel Support Vector Machine." Transportation Research Record: Journal of the Transportation Research Board 2324 (1): 44–52.

Chen, S., & Wang, W. (2009). "Decision Tree Learning for Freeway Automatic Incident Detection." Expert Systems with Applications 36 (2): 4101–4105.

Liu, Q., Jian Lu, & Chen., S. (2013). Traffic Incident Detection using Random Forest." Technical Report.

Li, L., He, S., Zhang, J., & Yang.., F. (2016). "Bagging-SVMs Algorithm-Based Traffic Incident Detection." CICTP 2016.

Ma, X., Yu, H., Wang, Y., & Wang., Y. (2015). *"Large-scale Transportation Network Congestion Evolution Prediction Using Deep Learning Theory." PloS One 10 (3.*

Lv, Y., Duan, Y., Kang, W., Li, Z., & Wang, F.-Y. (2015). *Traffic Flow Prediction with Big Data:ADeep Learning Approach, IEEE Transactionson Intelligent Transportation Systems 16 (2): 865–873.*

Ma, X., Dai, Z., He, Z., Ma, J., Wang, Y., & Wang, Y. (2017). *"Learning Traffic As Images: a Deep Convolutional Neural Network for Large-scale Transportation Network Speed Prediction." Sensors 17 (4): 818.*

Wang, X., & Abdel-Aty, M. (2006.). Temporal and spatial analyses of rear-end crashes at signalized intersections. Accident Analysis & Prevention, 38(6):1137–1150, .

Ren, H., Song, Y., Liu, J., Hu, Y., & Lei., J. (2017). A deep learning approach to the prediction of short-term traffic accident risk.

Chen, Q., Song, X., Yamada, H., & Shibasaki, R. (2016). Learning deep representation from big and heterogeneous data for traffic accident inference. In Thirtieth AAAI Conference on Artificial Intelligence,.

Li, Y., Yu, R., Shahabi, C., & Liu, Y. (2017). Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926}*.

Lin, L., Zhengbing, H., Srinivas, & Peeta. (2018). Predicting station-level hourly demand in a large-scale bike-sharing network: A graph convolutional neural network approach. *Transportation Research Part C: Emerging Technologies 97*, 258-276.

Cui, Z., Henrickson, K., Ke, R., & Wang, Y. (2019). Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting. *IEEE Transactions on Intelligent Transportation Systems, 21*(11), 4883-4894.

Cui, Z., Ke, R., Pu, Z., Ma, X., & Wang, Y. (2020). Learning traffic as a graph: A gated graph wavelet recurrent neural network for network-scale traffic prediction. . *Transportation Research Part C: Emerging Technologies,*, 115, 102620.

Farradyne, P. (2000). Traffic incident management handbook. Federal Highway Administration Office of Travel Management.

Nikolaev, A., Sapego, Y., Ivakhnenko, A., Mel'nikova, T., & Stroganov, V. (2017). Analysis of the incident detection technologies and algorithms in intelligent transport systems. International Journal of Applied Engineering Research, 15, 12. *Research India Publications, 4765-4774.*

Amini, S., Papapanagiotou, E., & Busch, F. (2016). Traffic management for major events. Digital Mobility Platforms and Ecosystems. State of the Art Report, TUM Living Lab Connected Mobility, 187- 197.

Beibei Ji, Y., Jiang, R., Qu, M., & Chung, E. (2014). Traffic incident clearance time and arrival time prediction based on hazard models. Mathematical Problems in Engineering, Hindawi Publishing Corporation,.

May., A. D. (1990). Traffic flow fundamentals. Transportation Research Board,.

Richards, P. I. (1956). Shock waves on the highway. Operations research, 4(1):42–51.

Francois Dion, H. R.-S. (2004). Comparison of delay estimates at under-saturated and over-saturated pre-timed signalized intersections. Transportation Research Part B: Methodological, 38(2):99–122.

Song, H., Triguero, I., & Ozcan, E. ⍰. (2019). A review on the self and dual interactions between machine learning and optimisation. *Progress in Artificial Intelligence* , 1-23.

Hutter, F., Kotthoff, L., & Vanschoren, J. (2019). Automated Machine Learning: Methods, Systems, Challenges. *Springer International Publishing, Cham*.

He, X., Zhao, K., & Chu, X. (2021). AutoML: A survey of the state-of-the-art. *Knowledge-Based Systems, 212*(106622).

Ratner, A., Bach, S. H., Ehrenberg, H., Fries, J., Wu, S., & Ré., C. (2020). Snorkel: Rapid training data creation with weak supervision. *VLDB J. 29, 2 (2020)*, 709–730.

Altman, N. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. . *Am. Stat., 46*((3)3), 175–185 .

Cortes, C., & Vapnik, V. (1995). Support-vector networks. . *Machine Learning, 20*(3), 273–297.

Thornton, C., Hutter, F., Hoos, H., & Leyton-Brown, K. (2013). Auto-WEKA. *. In: Proceedings of the 19th International conference on Knowledge discovery and data mining*, 847–855.

Komer, B., Bergstra, J., & Eliasmith, C. (2014). Hyperopt-sklearn: automatic hyperparameter configuration for scikit-learn. *Proceedings of SciPy*, 33–39.

Feurer, M., Klein, A., Eggensperger, K., Springenberg, J., Blum, M., & Hutter, F. (2015). Efficient and robust automated machine learning. *Advances in Neural Information Processing Systems, 28*, 2962–2970 .

Sparks, E., Talwalkar, A., Haas, D., Franklin, M., Jordan, M., & Kraska, T. (2015). Automating model search for large scale machine learning. *Proceedings of System-on-Chip Conference*, 368–380.

Sabharwal, A., Samulowitz, H., & Tesauro, G. (2016). Selecting near-optimal learners via incremental data allocation. *Proceedings of the Thirtieth Conference on Artificial Intelligence*, 2007–2015 .

Olson, R., Bartley, N., Urbanowicz, R., & Moore, J. (2016). Evaluation of a tree-based pipeline optimization tool for automating data science. *Proceedings of the Genetic and Evolutionary Computation Conference 2016* (pp. 485–492.). New York: ACM.

Swearingen, T., Drevo, W., Cyphers, B., Cuesta-Infante, A., Ross, A., & Veeramachaneni, K. (2017). ATM: a distributed, collaborative, scalable system for automated machine learning. *2017 IEEE International Conference on Big Data*, 151–162.

Mohr, F., Wever, M., & Hüllermeier, E. (2018). ML-Plan: automated machine learning via hierarchical planning. . *Mach. Learn., 107*(8), 1495–1515.

Jin, H., Song, Q., & Hu, X. ( 2019). Auto-keras: an efficient neural architecture search system. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining,*, 1946–1956.

Nikitin, N. e. (2022). Automated evolutionary approach for the design of composite machine learning pipelines. . *Future Gener. Comput. Syst. 127*, 109–125.

Gijsbers, P., LeDell, E., Thomas, J., Poirier, S., Bischl, B., & Vanschoren, J. (2019). An open source AutoML benchmark. *arXiv*.

Zöller, M., & Huber, M. (2021). Benchmark and survey of automated machine learning frameworks. . *J. Artif. Intell. Res. 70*, 409–472 .

Fikardos, M., Lepenioti, K., Bousdekis, A., Bosani, E., Apostolou, D., & Mentzas, G. (2022). An Automated Machine Learning Framework for Predictive Analytics in Quality Control. . *Advances in Production Management Systems. Smart Manufacturing and Logistics Systems: Turning Ideas into Action. APMS 2022. IFIP Advances in Information and Communication Technology,, 663*.

Angarita-Zapata, J., Masegosa, A., & Triguero, I. (2020). Evaluating automated machine learning on supervised regression traffic forecasting problems. *Computational Intelligence in Emerging Technologies for Engineering Applications. , 872*, 187–204.

Angarita-Zapata, J., Triguero, I., & Masegosa, A. (2018). A preliminary study on automatic algorithm selection for short-term traffic forecasting. *IDC 2018, 798*, 204–214.

Vlahogianni, E. (2015). Optimization of traffic forecasting: intelligent surrogate modeling. *Transp. Res. Part C Emerg. Technol. 55*, 14–23.

Angarita-Zapata, J., Masegosa, A., & Triguero, I. (2020). General-Purpose Automated Machine Learning for Transportation: A Case Study of Auto-sklearn for Traffic Forecasting. . *Information Processing and Management of Uncertainty in Knowledge-Based Systems, 1238*.

Hutter, F., Hoos, H., & Leyton-Brown, K. (2011). Sequential model-based optimization for general algorithm configuration. *Coello, C.A.C. (ed.) LION 2011. LNCS, 6683*, 507–523.

Qian, G., & Liu, J. (2022). A comparative study of deep learning-based fault diagnosis methods for rotating machines in nuclear power plants. . *Ann. Nucl. Energy, 178*(109334).

Gain, U., & Hotti, V. (2021). Low-code AutoML-augmented data pipeline–a review and experiments. . *J. Phys. Conf. Ser. , 1828*(012015).

Le, T. T., Fu, W., & Moore, J. H. (2020). Scaling tree-based automated machine learning to biomedical big data with a feature set selector. . *Bioinformatics.36(1)*, 250-256.

Kashinath, S., Mostafa, S., Mustapha, A., Mahdin, H., Lim, D. &., Mohammed, M., . . . Yang, T. (2021). Review of data fusion methods for Real-time and Multi-Sensor Traffic Flow Analysis. *IEEE Access*.

Aimsun. (2023). Aimsun Next 23 User's Manual, Aimsun Next Version 23.0.0. Barcelona, Spain.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique.

Tomek, I. (1976). Two Modifications of CNN. IEEE Transactions on Systems, Man, and Cybernetics, vol. 6, no. 11, pp. 769–772.

Zeng, M., Zou, B., Wei, F., Liu, X., & Wang, L. (2016). Effective prediction of three common diseases by combining SMOTE with Tomek links technique for imbalanced medical data.

Swana, E. F., Doorsamy, W., & Bokoro, P. (2022). settings Open AccessArticle Tomek Link and SMOTE Approaches for Machine Fault Classification with an Imbalanced Dataset.

Simeone, O. (2018). A Brief Introduction to Machine Learning for Engineers. In *Trends®Signal Process.* (pp. 200–431).

Zhou, J., Gandomi, A., Chen, F., & Holzinger, A. (2021). Evaluating the Quality of Machine Learning Explanations: A Survey on Methods and Metrics. *Electronics, 10*(5), 593.

Ghosh, B., & Smith, D. P. (2014). Customization of automatic incident detection algorithms for signalized urban arterials. *Journal of Intelligent Transportation Systems, 18*(4), 426-441.

Torrent-Fontbona, F., Dominguez, M., Fernandez, J., & Casas, J. (2023). Towards Efficient Incident Detection in Real-time Traffic Management. *4th Symposium on Management of Future Motorway and Urban Traffic Systems (MFTS).* Dresden.

Herrmann, M., & Scheipl, F. (2021). A Geometric Perspective on Functional Outlier Detection. *Stats, 4*(4), 971-1011.

Yuan, F., & Cheu, R. L. (2003). Incident detection using support vector machines, Transportation Research Part C: Emerging Technologies, Volume 11, Issues 3–4, Pages 309-328,.

Kopčan, J., Škvarek, O., & Klimo, M. (2021). Anomaly detection using Autoencoders and Deep Convolution Generative Adversarial Networks. *Transportation Research Procedia*(55), 1296-1303.

Ashraf, J., Bakhshi, A. D., Moustafa, N., Khurshid, H., Javed, A., & Beheshti, A. (2020). Novel deep learning-enabled LSTM autoencoder architecture for discovering anomalous events from intelligent transportation systems. . *IEEE Transactions on Intelligent Transportation Systems, 22*(7), 4507-4518.

ElSahly, O., & Abdelfatah, A. (2022). Systematic Review of Traffic Incident Detection Algorithms. *Sustainability, 14*, 14859.

Karmaker Santu, S. K., Mahadi, H. M., Smith, M. J., Xu, L., Zhai, C., & Veeramachaneni, K. (2022). AutoML to Date and Beyond: Challenges and Opportunities. *ACM Comput. Surv. 54, 8*(Article 175).

ENISA - European Union Agency for Cybersecurity. (2020). *AI CYBERSECURITY CHALLENGES.* ENISA.

Huang, Z., Xu, W., & Yu., K. (2015). Bidirectional LSTM-CRF models for sequence tagging.

Jiang, W., & Luo, J. (2022). Graph neural network for traffic forecasting: A survey,. *Expert Systems with Applications, 207*, 117921.

Li, L., Sheng, X., Du, B., Wang, Y., & Ran., B. (2020). “A Deep Fusion Model Based on Restricted Boltzmann Machines for Traffic Accident Duration Prediction.” Engineering Applications of Artificial Intelligence 93.

FIWARE, (. (n.d.). *Orion Context Broker.* Retrieved from https://fiware-orion.readthedocs.io/en/master/

Gutiérrez, V., Martínez, V., & Sánchez, L. (2019). Middleware architectures for scalable IoT systems: The FIWARE case. Journal of Ambient Intelligence and Smart Environments. *11*(4), 387-408.

Wang, X., & Chen, J. (2018). The role of context brokers in IoT applications: A survey. . *Internet of Things*, 18-28.

Gyrard, A., Serrano, M., & Atemezing, G. A. (2017). Developing smart city applications using the FIWARE platform: A case study. *Procedia Computer Science*(110), 340-347.

Smart Data Models. (n.d.). *The Smart Data Models Initiative Embraces JSON Schema as the Core Component for Interoperable Smart Solutions.* Retrieved from Smart Data Models: https://smartdatamodels.org/

Olugbade, S., Ojo, S., Imoize, A., Isabona, J., & Alaba, M. (2022). A Review of Artificial Intelligence and Machine Learning for Incident Detectors in Road Transport Systems. *Math. Comput. Appl., 27*(5), 77.

Liang, H., Song, H., Yun, X., Sun, S., Wang, Y., & Zhang, Z. (2022). Traffic incident detection based on a global trajectory spatiotemporal map. *Complex Intell. Syst., 8*, 1389–1408 .

Adadi, A., & Bouhoute, A. (2023). *Explainable Artificial Intelligence for Intelligent Transportation Systems.* CRC Press.

Sahoo, L. G., & Mohan, B. (2022). Introduction to Explainable AI and Intelligent Transportation. In *Explainable Artificial Intelligence for Intelligent Transportation Systems.*

Minh, D., Wang, H., & Li, Y. e. (2022). Explainable artificial intelligence: a comprehensive review. *Artif Intell Rev* , 3503–3568.

Ali, S., Abuhmed, T., El-Sappagh, S., Muhammad, K., Alonso-Moral, J. M., Confalonieri, R., . . . Francisco Herrera. (2023). Explainable Artificial Intelligence (XAI): What we know and what is left to attain Trustworthy Artificial Intelligence. *Information Fusion*, 1566-2535.

Rieg, T., Frick, J., Baumgartl, H., & Buettner, R. (2020). Demonstration of the potential of white-box machine learning approaches to gain insights from cardiovascular disease electrocardiograms. *PLoS One 15*.

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., & Pedreschi, D. (2018). A survey of methods for explaining black box models. *ACM Comput. Surv. 51 (5)* , 1–42.

Véliz, C., Prunkl, C., Phillips-Brown, M., & Lechterman, T. (2021). We might be afraid of black-box algorithms. *J. Med. Ethics 47* , 339–340.

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2021). A survey on bias and fairness in machine learning. *ACM Comput. Surv. 54* , 1–35.

Oberman, C. F. (2021). Scaleable input gradient regularization for adversarial robustness. *Mach. Learn. Appl. 3* .

Li, X., Xiong, H., Li, X., Wu, X., Zhang, X., Liu, J., . . . Dou, D. (2022). Interpretable deep learning: interpretation, interpretability, trustworthiness, and beyond, . *Knowledge and Information Systems 64 (12)* , 3197–3234.

Das, A., & Rad, P. (2020). Opportunities and challenges Explainable Artificial Intelligence (XAI): A survey.

Ribeiro, M., Singh, S., & Guestrin, C. (2016). "Why should i trust you?" explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (pp. 1135–1144).

Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. *ArXiv*.

Ahsan, M. M., & Siddique, Z. (2021). Machine learning based disease diagnosis: A comprehensive review. .

Kumar, S., Datta, S., Singh, V., Datta, D., Sing, S. K., & Sharma, R. (2024). Applications, Challenges, and Future Directions of Human-in-the-Loop Learning. *IEEE Access*, 75735-75760.

Gupta, R., Gupta, H., & Mohania, M. (2012). Cloud computing and big data analytics: what is new from databases perspective?. In *International conference on big data analytics* (pp. 42-61). Berlin, Heidelberg: Springer.

Han, J., Haihong, E., Le, G., & Du, J. (2011). Survey on NoSQL database. *2011 6th international conference on pervasive computing and applications* (pp. 363-366). IEEE.

Konstantinou, I., Angelou, E., Boumpouka, C., Tsoumakos, D., & Koziris, N. (2011). On the elasticity of nosql databases over cloud management platforms. *Proceedings of the 20th ACM international conference on Information and knowledge management*, (pp. 2385-2388).

Liu, W., Zheng, Y., Chawla, S., Yuan, J., & Xie, X. (2011). Discovering spatio-temporal causal interactions in traffic data streams. . *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (pp. 1010–1018).

Nguyen, H., Liu, W., & Chen, F. (2016). Discovering Congestion Propagation Patterns in Spatio-Temporal Traffic Data. *IEEE Transactions on Big Data, 3*(2), 169–180.

Liang, Y., Jiang, Z., & Zheng, Y. (2017). Inferring Traffic Cascading Patterns. . *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems.*

He, Y., Wang, L., Fang, Y., & Li, Y. (2018). Discovering Congestion Propagation Patterns by Co-location Pattern Mining. . *Lecture Notes in Computer Science (Including Subseries 103 Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics*, (pp. 46-55).

Wang, L., & Zhou, Z. (2017). Congestion Prediction for Urban Areas by Spatiotemporal Data Mining. *Proceedings - 2017 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery* (pp. 290-297). CyberC.

Ji, X., Wang, J. S., Zhou, T. M., & Chen, P. J. (2019). Recurrent city traffic congestion propagation analysis. *Journal of Physics: Conference Series, 1168*(3), 32093.

Xiong, H., Vahedian, A., Zhou, X., Li, Y., & Luo, J. (2018). Predicting traffic congestion propagation patterns: A propagation graph approach. IWCTS 2018. - *Proceedings of the 11th ACM SIGSPATIAL International Workshop on Computational Transportation Science*, *18*, pp. 60-69.

Majumdar, S., Subhani, M. M., Roullier, B., Anjum, A., & Zhu, R. (2021). Congestion prediction for smart sustainable cities using IoT and machine learning approaches. *Sustainable Cities and Society, 64*, 102500.

Wu, J., Huang, W., Boer, N. d., Mo, Y., He, X., & Lv, C. (2022). Safe decision-making for lane-change of autonomous vehicles via human demonstration-aided reinforcement learning,. *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, (pp. 1228-1233).

Holmberg, L., Davidsson, P., & Linde, P. (2020). A feature space focus in machine teaching. *2020 IEEE international conference on pervasive computing and communications workshops (PerCom Workshops)*, (pp. 1-2).

Settles, B. (2009). Active learning literature survey. *Tech. rep., University of Wisconsin-Madison. Department of Computer Sciences,*.

Amershi, S., Cakmak, M., & Knox, W. e. (2014). Power to the people: the role of humans in interactive machine learning. *AI Magazine, 35*(4), 105-120.

Simard, P., Amershi, S., & Chickering, D. e. (2017). Machine teaching: A new paradigm for building machine learning systems.

Liu, Z., Feng, X., & Wang, Y. e. (2021). Self-paced learning enhanced neural matrix factorization for noise-aware recommendation. *Knowl Based Syst, 213*(106), 660.

Arous, I., Dolamic, L., Yang, J., A.Bhardwaj, Cuccu, G., & CudreI-Mauroux, P. (2021). Marta: Leveraging human rationales for explainable text classification,. *The AAAI Conference on Artificial Intelligence, 35*(7), 5868–5876.

Madono, K., Nakano, T., Kobayashi, T., & Ogawa, T. (2020). Efficient human-in-the-loop object detection using bi-directional deep sort and annotation-free segment identification. *APSIPA ASC*, 1226–1233.

Sing, H. V., & Mahmoud, Q. H. (2020). Human-in-the-loop error precursor detection using language translation modeling of hmi states,. *SMC*, 2237-2242.

Machiry, A., Tahiliani, R., & Naik, M. (2013). Dynodroid: An input generation system for android apps,. *Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering*, (pp. 224-234).

Demirel, ]. H. (2020). Digital human-in-the-loop framework. *International Conference on Human-Computer Interaction*, (pp. 18-32).

Zhang, S., He, L., Dragut, E., & Vucetic, S. (2019). How to invest mytime: Lessons from human-in-the-loop entity extraction. *KDD*, 2305-2313.

Chiang, H. -H., Wu, S. -J., Perng, J. -W., Wu, B. -F., & Lee, T. -T. (2010). The Human-in-the-Loop Design Approach to the Longitudinal Automation System for an Intelligent Vehicle. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 40*(4), 708-720.

Agnisarman, S., Lopes, S., Madathil, K. C., Piratla, K., & Gramopadhye, A. (2019). A survey of automation-enabled human-in-the-loop systems for infrastructure visual inspection. *Automation in Construction, 97*, 52-76.

Huang, Z., Sheng, Z., Ma, C., & Sikai Chen. (2024). Human as AI mentor: Enhanced human-in-the-loop reinforcement learning for safe and efficient autonomous driving. *Communications in Transportation Research, 4*, 100127.

INRIX. n.d. (2024, 9 25). Retrieved from 2022 Global Traffic Scorecard.: https://inrix.com/scorecard/

Mokbel, M., Xiong, M. S., Züfle, A., Almeida, J., Anderson, T., Aref, W., . . . Gunopu, D. (2024). Mobility Data Science: Perspectives and Challenges. *ACM Trans. Spatial Algorithms Syst.* , Article 10, 35 pages.

Li, X., Han, J., Lee, J.-G., & Gonzalez, H. (2007). Traffic density-based discovery of hot routes in road networks. *Proceedings of Advances in Spatial and Temporal Databases: 10th International Symposium, SSTD 2007,* (pp. 441-459). Boston, MA: Springer.

Maeda, H., Sekimoto, Y., & Seto, T. (2016). Lightweight road manager: Smartphone-based automatic determination of road damage status by deep neural network. *Proceedings of the 5th ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems.*, (pp. 37-45).

Li, Y., Yu, R., Shahabii, C., & Liu, Y. (2018). Diffusion convolutional recurrent neural network: Datadriven traffic forecasting. *Conference Track Proceedings of 6th International Conference on Learning Representations, ICLR 2018.* Vancouver, BC, Canada.

Zhang, P., Zhang, H., & Guo, D. (n.d.). Evacuation shelter and route selection based on multi-objective optimization approach. *Proceedings of the 1st ACM SIGSPATIAL InternationalWorkshop on the Use of GIS in Emergency Management*, (pp. 1-5).

Richly, K., Teusner, R., Immer, A., Windheuser, F., & Wolf, L. (2015). Optimizing routes of public transportation systems by analyzing the data of taxi rides. *Proceedings of the 1st International ACM SIGSPATIAL Workshop on Smart Cities and Urban Analytics*, (pp. 70-76).

Souza, A. M., Yokoyama, R. S., Maia, G., Loureiro, A., & Villas, L. (2016). Real-time path planning to prevent traffic jam through an intelligent transportation system. *2016 IEEE Symposium on Computers and Communication (ISCC).* (pp. 726-731). IEEE.

Kolahdouzan, M. R., & Shahabi, C. (2004). Voronoi-based K nearest neighbor search for spatial network databases. In M. A. Nascimento, M. T. Özsu, D. Kossmann, R. J. Miller, '. A. Blakeley, & K. B. Schiefer, *e)Proceedings of the 30th*

*International Conference on Very Large Data Bases, VLDB 2004,* (pp. 840–851). Toronto, Canada: Morgan Kaufmann.

Li, Y., Gunopulos, D., Lu, C., & Guibas, L. J. (2019). Personalized travel time prediction using a small number of probe vehicles. *ACM Trans. Spatial Algorithms Syst. 5, 1*, 4:1–4:27.

Lin, C., Choy, K. L., Ho, G. T., Sai Ho Chung, & Lam, H. Y. (2014). Survey of green vehicle routing problem: Past and future trends. *Expert Systems with Applications, 41*(4), 1118–1138.

Tomaras, D., Kalogeraki, V., Liebig, T., & Gunopulos, D. (2018). Crowd-based ecofriendly trip planning. *19th IEEE International Conference on Mobile Data Management, MDM 2018* (pp. 24-33). Aalborg, Denmark: IEEE Computer Society.

Ahmed, M., Karagiorgou, S., Pfoser, D., & Wenk, C. (2015). *Map Construction Algorithms.* Springer.

Chen, C., Lu, C., Huang, Q., Yang, Q., Gunopulos, D., & Guibas, L. J. (2016). City-scale map creation and updating using GPS collections. In B. Krishnapuram, M. Shah, A. J. Smola, C. C. Aggarwal, D. Shen, & R. Rastogi (Ed.), *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* (pp. 1465–1474). San Francisco, CA,: ACM,.

Macfarlane, J., & Stroila, M. (2016). Addressing the uncertainties in autonomous driving. *SIGSPATIAL Special 8*(2), 35-40.

Shao, W., Prabowo, A., Zhao, S., Tan, S., Koniusz, P., Chan, J., . . . Salim, F. D. (2019). Flight delay prediction using airport situational awareness map. *Proceedings* *of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, (pp. 432–435).

Zhao, K., Tarkoma, S., Liu, S., & Vo, H. (2016). Urban human mobility data mining: An overview. *2016 IEEE International Conference on Big Data (Big Data).* (pp. 1911–1920). IEEE.

Claramunt, C., Ray, C., Salmon, L., Camossi, E., Hadzagic, M., Jousselme, A. L., . . . Vouros, G. (2017). Maritime data integration and analysis: recent progress and research challenges. . *Advances in Database Technology-EDBT*, 192–197.

Chung, S.-H., Ma, H.-L., Mark Hansen, & Choi, T.-M. (2020). Data science and analytics in aviation. 101837.

Ossi, F., Hachem, F., Cagnacci, F., Demšar, U., & Damiani., M. L. (2022). HaniMob 2021 Workshop Report: The 1st ACM SIGSPATIAL Workshop on Animal Movement Ecology and Human Mobility. *SIGSPATIAL Special 13, 1-3* , 33–36.

Jensen, C. S., Lu, H., & Yang, B. (2010). Indoor—A new data management frontier. *IEEE Data Eng. Bull. 33, 2* , 12–17.

Pan, B., Zheng, Y., Wilkie, D., & Shahabi., C. (2013). Crowd sensing of traffic anomalies based on human mobility and social media. *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems.* , (pp. 344–353).

Nikitopoulos, P., Paraskevopoulos, A.-I., Doulkeridis, C., Pelekis, N., & Theodoridis., Y. (2018). Hot spot analysis over big trajectory data. *2018 IEEE International Conference on Big Data (Big Data).*, (pp. 761–770).

Nag, A. M., & Simon, V. (2018). Survey on traffic prediction in smart cities. . *Pervasive and Mobile Computing 50* , 148–163.

Wang, H., Tang, X., Kuo, Y.-H., Kifer, D., & Li, Z. (2019). A simple baseline for travel time estimation using large-scale trip data. *ACM Transactions on Intelligent Systems and Technology (TIST) , 10*(2), 1-22.

Wang, S., Bao, Z., Culpeppe, J. S., & Cong., G. (2021). A survey on trajectory data management, analytics, and learning. *ACM Comput. Surv., 54*(2), 36 pages.

Toohey, K., & Duckham., M. (2015). Trajectory similarity measures. *SIGSPATIAL Special, 7*(1), 43–50.

Han, X., Cheng, R., Ma, C., & Grubenmann, T. (2022). DeepTEA: Effective and efficient online time-dependent trajectory outlier detection. *Proceedings of the VLDB Endowment* , *15, 7*, pp. 1493–1505.

Biljecki, F., Ledoux, H., & Oosterom, P. V. (2013). Transportation mode-based segmentation and classification of movement trajectories. *International Journal of Geographical Information Science, 27*(2), 385–407.

Sakr, M. A., & Güting, R. H. (2014). Group spatiotemporal pattern queries. . *GeoInformatica 18* , 699–746.

Krumm., J. (2022. ). Maximum entropy bridgelets for trajectory completion. . *Proceedings of the 30th International Conference on Advances in Geographic Information Systems.* , 1–8.

Joo, R., Boone, M. E., Clay, T. A., Patrick, S. C., Clusella-Trullas, S., & Basille, M. (2020). Navigating through the R packages for movement. . *Journal of Animal Ecology 89, 1 (Jan. 2020)*, 248–267.

Graser., A. (2023). The state of trajectory visualization in notebook environments. *GI_Forum 1*, 73–91.

Gao, N., Xue, H., Shao, W., Zhao, S., Qin, K. K., Prabowo, A., . . . Salim., F. D. (2022). Generative adversarial networks for spatio-temporal data: A survey. . *ACM Transactions on Intelligent Systems and Technology (TIST) 13,* .

Xue, H., & Salim, F. D. (2021). TERMCast: Temporal relation modeling for effective urban flow forecasting. *Advances in Knowledge Discovery and Data Mining: 25th Pacific-Asia Conference, PAKDD 2021.* (pp. 741–753). Springer.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., . . . Polosukhin., I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems 30* .

Wang, Z., Jiang, R., Xue, H., Salim, F. D., Song, X., & Shibasaki, R. (2022). Event-aware multimodal mobility nowcasting. *Proceedings of the AAAI Conference on Artificial Intelligence, 36*, 4228–4236.

Zhao, S., Shao, W., Chan, J., & Salim, F. D. (2022). Measuring disentangled generative spatio-temporal representation. . *Proceedings of the 2022 SIAM International Conference on Data Mining (SDM). SIAM*, 522–530.

Lee, C., Kim, Y., Jin, S., Kim, D., Maciejewski, R., Eber, D., & Ko, S. (2020). A visual analytics system for exploring, monitoring, and forecasting road traffic congestion. *IEEE Transactions on Visualization and Computer Graphics, 26*(11), 3133-3146.

Andrienko, G., Andrienko, N., Bak, P., Keim, D., & Wrobel, S. (2013). *Visual Analytics of Movement.* Springer.

Almohammad, A., & Georgakis, P. (2023). Automated Approach for Generating and Evaluating Traffic Incident Response Plans. *Eng. Proc., 39*(1).

Almohammad, A., & Georgakis, P. (2022). Machine Learning Based Method for Modeling Traffic Events. *2022 3rd International Conference on Computation, Automation and Knowledge Management (ICCAKM).* Dubai, United Arab Emirates: IEEE.

Jones, A. S., Georgakis, P., Petalas, Y., & Suresh, R. (2018). Real-time traffic event detection using Twitter data. *Infrastructure Asset Management, 5*(3), 77-84.

Almohammad, A., & Georgakis, P. (2020). Public Twitter Data and Transport Network Status. *2020 10th International Conference on Information Science and Technology (ICIST).* Bath, London, and Plymouth, UK: IEEE.

Gkioka, G., Onrubia, J., Dominguez, M., Zormpas, S., Magoutas, B., Mroz, A., & Torres, P. (2024). AI and data-driven approach for next generation traffic management, Lecture Notes in Mobility, . *Transport Research Arena 2024.* Dublin, Ireland.

Bui, K.-H. N., Cho, J., & Yi, H. (2021). Spatial-temporal graph neural network for traffic forecasting: An overview and open research issues. *Springer Nature Applied Intelligence*, 2763–2774.

Celesti, A., Fazio, M., Márquez, F. G., Glikson, '., Mauwa, H., Bagula, A., . . . Villari, M. (2019). How to Develop IoT Cloud e-Health Systems Basedon FIWARE: A Lesson Learnt. *Journal of Sensor and Actuator Networks, 8*(1).

Payne, H. J., & Tignor, S. C. (1978). Freeway incident-detection algorithms based on decision trees with states. *Transportation Research Record (682)*, 30–37.

Ahmed, S., & Cook, A. (1982). Application of Time-Series analysis techniques to freeway incident detection. *Transportation Research Record* , 19-21.

Agarwal, S., Kachroo, P., & Regentova, E. (2016). A hybrid model using logistic regression and wavelet transformation to detect traffic incidents. *IATSS Research*, 56-63.

Lin, H., Liu, Y., Li, S., & Qu, X. (2023). How Generative Adversarial Networks Promote the Development of Intelligent Transportation Systems: A Survey. *IEEE/CAA Journal of Automatica Sinica*.

Ahmed, M., & Cook, A. (1979). ANALYSIS OF FREEWAY TRAFFIC TIME-SERIES DATA BY USING BOX-JENKINS TECHNIQUES. *722*.

Chassiakos, A., & Stephanedes, Y. (1993). Smoothing algorithms for incident detection. *Transp. Res. Rec.* , 8-16.

Wang, R., Fan, S., & Work., D. B. (2016). Efficient multiple model particle filtering for joint traffic state estimation and incident detection. *Transportation Research Part C: Emerging Technologies 71* , 521-537.

Ki, Y. -K., Jeong, W. -T., Kwo, H. -J., & Kim, M. -R. (2019). An Algorithm for Incident Detection Using Artificial Neural Networks. *2019 25th Conference of Open Innovations Association (FRUCT)*, (pp. 162-167). Helsinki, Finland.

Ansari Esfe, R. (2021). *https://prism.ucalgary.ca.* Retrieved from Urban Traffic Congestion Propagation Prediction Model: A case of non-recurrent congestion: https://prism.ucalgary.ca/server/api/core/bitstreams/a8d0487e-6529-4fcd-baad-69743e776fef/content

Yuan, X., Li, L., & Wang, Y. (2019). Nonlinear Dynamic Soft Sensor Modeling With Supervised Long Short-Term Memory Network. *IEEE Transactions on Industrial Informatics*.

Plevris, V., Solorzano, G., Bakas, N. P., & Ben Seghier, M. E. (2022). Investigation of performance metrics in regression analysis and machine learning-based prediction models. *ECCOMAS Congress;8th European Congress on Computational Methods in Applied Sciences and Engineering.* European Community on Computational Methods in Applied Sciences .

Munro, R. (2020). *Human-in-the-loop machine learning.* Shelter Island: Manning Publications.

Torrent-Fontbona, F., Dominguez, M., Fernandez, J., & Casas, J. (2022). Towards Efficient Incident Detection in Real-time Traffic Management. *4th Symposium on Management of Future Motorway and Urban Traffic Systems (MFTS). .* Dresden.

# List of Publications

**Journal Publications**

[j1] **Gkioka, G**., Bothos, T., Magoutas, B., & Mentzas, G. (2023). Data analytics methods to measure service quality: A systematic review. Intelligent Decision Technologies, IOS Press, November 2023. DOI: 10.3233/idt-230363.

[j2] **Gkioka, G.**, Dominguez, M., & Mentzas, G. (2024). An AutoML-based approach for automatic traffic incident detection in smart cities. Intelligent Decision Technologies, Pre-press, 1-22. DOI: 10.3233/IDT-240231.

[j3] **Gkioka, G**., Dominguez, M., Tympakianaki, A., Mentzas, G. (2024) AI-Driven Real-Time Incident Detection for Intelligent Transportation Systems, Advances in Transdisciplinary Engineering, Pages 56 - 68, DOI: 10.3233/ATDE240021.

[j4] Koronakos, G., **Gkioka, G.,** Mentzas, G. Mathematical Programming models to Recommend Response Plans for Traffic Incident Management, Operational Research - *Under Review*

**Conference Publications – Proceedings**

[c1] **Gkioka, G**., Mentzas, G., Human-Centric and Explainable AI for Real-Time Traffic Incident Detection in Smart Transportation Systems, Navigating the Future of Traffic Management International Symposium, Athens, Greece - *Under Review*

[c2] **Gkioka, G.,** Onrubia, J., Dominguez, M., Zormpas, S., Magoutas, B., Mroz, A., & Torres, P.(2024). AI and data-driven approach for next generation traffic management, Lecture Notes in Mobility, Transport Research Arena 2024, Dublin, Ireland.

[c3] **Gkioka, G.,** Dominguez, M., & Mentzas, G. (2023). Automatic incident detection with AI-based methods using heterogeneous multimodal big data. *ISTDM 2023*, Ispra, Italy.

[c4] **Gkioka, G**., & Mentzas, G. (2023). Data-driven incident detection in intelligent transportation systems. *5th Summit of Gender Equality in Computing*, Athens, Greece.

[c5] Kalfa, N., **Gkioka, G**., & Torres Álvarez, P. (2023). FRONTIER EU Project: Next generation traffic management for empowering CAVs integration, cross-stakeholders collaboration and proactive multi-modal network optimization. *ITS2023: Intelligent Systems and Consciousness Society*, Patras, Greece.

[c6] **Gkioka, G**., Magoutas, B., Bothos, E., & Mentzas, G. (2022). A hybrid data model for the assessment of border control technologies. In *2022 13th International Conference on Information, Intelligence, Systems & Applications (IISA)* (pp. 1-8). IEEE. DOI: 10.1109/IISA56318.2022.9904400.

[c7] **Gkioka, G.** (2022). A data analytics framework for monitoring technology acceptance of SBC technologies. *9th EAB Research Projects Conference 2022*.

[c8] **Gkioka, G.**, & Mentzas, G. (2022). Data analytics for technology acceptance monitoring of SBC technologies. Special Track 'The future of border and external security: From data to policies' at the *Data for Policy 2022: Ecosystems of Innovation and Virtual-Physical Interactions*.

[c9] Rocha, A., Blanchard, S., Fraga, J., **Gkioka, G**., Gomes, P., Gonzalez, L., Krastev, T., Riddone, G., & Widegren, D. (2017). Integration of the vacuum SCADA with CERN's Enterprise Asset Management system. *16th International Conference on Accelerator and Large Experimental Physics Control Systems (ICALEPCS 2017)*, Barcelona, Spain, 8-13 October 2017, pp. TUPHA044.

# Resume

Mrs. Georgia Gkioka is a PhD candidate at the National Technical University of Athens (NTUA), specializing in Artificial Intelligence for Transportation. Her doctoral research focuses on "Human-in-the-Loop Predictive Analytics for Incident Detection in Smart Transportation Systems," a topic at the intersection of machine learning, explainability, and intelligent transport systems. She holds a Master of Engineering from NTUA's School of Electrical and Computer Engineering, where she graduated with distinction (GPA 8.10/10, while she entered12th in her class with 19,763 points). She is a member of the Technical Chamber of Greece (TEE) and owner of professional permits in Electrical and Computer Engineering from 2018. She holds the Certificate of Proficiency in English provided by the University of Cambridge and Michigan (proficient knowledge of English) and C1 DALF Diplôme approfondi de langue française (proficient knowledge of French). She has used both English and French in the professional context while living abroad in Geneva, Switzerland and Brussels, Belgium.

Since 2020, Mrs. Gkioka has been working as a researcher at the Information Management Unit (IMU) of the Institute of Communication and Computer Systems (ICCS) of the National Technical University of Athens (NTUA), where she has participated in three (3) research and innovation projects. Her technical expertise includes data analytics and machine learning, Automated Machine Learning, explainability in AI, with applications in practical, real-world projects. Prior to this, she worked as a Technical and Business Analyst at STIB-MIVB in Brussels (2018-2020), where she led DevOps transformations using CI/CD, container orchestration, cloud platforms, and configuration automation. Her consulting role at ADNEOM in Brussels provided experience in implementing data analytics and business intelligence solutions, while her early career included roles at CERN, NOKIA, and the Council of the EU mainly as software engineer.

Mrs. Gkioka has also been active as a certified Soft Skills Trainer since 2014, delivering over 140 hours of training on topics such as presentation skills, project management, and leadership. She has been volunteering in various initiatives, which include organizing the 1st Athens Soft Skills Academy and coordinating a Job Fair at the Council of the EU, focusing on empowering young professionals. She is a passionate advocate for women in science and technology sectors, having volunteered with the Womenpreneur Initiative in Brussels and having chaired EESTEC LC Athens, while currently being a mentor for the Greek Women in STEM organization.

Since September 2020, Mrs. Gkioka has been a PhD candidate at the Information Management Unit of the School of Electrical and Computer Engineering of NTUA. Her research interests encompass AI-driven predictive analytics and explainability techniques, particularly for applications in transportation.