



NATIONAL TECHNICAL UNIVERSITY OF ATHENS
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING
SCHOOL OF MECHANICAL ENGINEERING

INTERDISCIPLINARY POSTGRADUATE PROGRAMME
“Translational Engineering in Health and Medicine”

***Machine Learning Algorithms for Detecting Fatigue:
An EEG Data Analysis***

Postgraduate Diploma Thesis

MICHAIL N. SOFRAS

Supervisor: Dr. George Matsopoulos
Professor in School of Electrical and Computer Engineering
National Technical University of Athens

Athens, (February 2025)



NATIONAL TECHNICAL UNIVERSITY OF ATHENS
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING
SCHOOL OF MECHANICAL ENGINEERING

INTERDISCIPLINARY POSTGRADUATE PROGRAMME
“Translational Engineering in Health and Medicine”

***Machine Learning Algorithms for Detecting Fatigue:
An EEG Data Analysis***

Postgraduate Diploma Thesis

MICHAIL N. SOFRAS

Supervisor: Dr. George Matsopoulos
Professor in School of Electrical and Computer Engineering
National Technical University of Athens

The postgraduate diploma thesis has been approved by the examination committee
on 12th February 2025

1st member

2nd member

3rd member

Dr George Matsopoulos

Dr Panayiotis Tsanakas

Dr Christos Manopoulos

*Professor in School of
Electrical and Computer
Engineering
National Technical
University of Athens)*

*Professor of Computer
Engineering
National Technical
University of Athens*

*Assistant Professor in School
of Mechanical Engineering
National Technical
University of Athens*

Athens, (February 2025)

.....
Michail N. Sofras
Graduate of the Interdisciplinary Postgraduate Programme,
“Translational Engineering in Health and Medicine”,
Master of Science,
School of Electrical and Computer Engineering,
National Technical University of Athens

Copyright © - (*Michail N. Sofras 2025*)
All rights reserved.

You may not copy, reproduce, distribute, publish, display, modify, create derivative works, transmit, or in any way exploit this thesis or part of it for commercial purposes. You may reproduce, store or distribute this thesis for non-profit educational or research purposes, provided that the source is cited, and the present copyright notice is retained. Inquiries for commercial use should be addressed to the original author.

The ideas and conclusions presented in this paper are the author’s and do not necessarily reflect the official views of the National Technical University of Athens.

Abstract

Mental fatigue considerably affects cognitive performance, decision-making, and general productivity in diverse fields such as healthcare, transportation, and military operations. Extended cognitive strain may result in diminished vigilance and increasing error rates, presenting significant hazards to safety and productivity. Electroencephalography (EEG), a non-invasive technique for monitoring cerebral activity, offers an objective approach to identify and categorize mental fatigue. This thesis introduces a robust machine learning framework for classifying EEG data into rest and fatigue states, highlighting the application of specialized feature selection methods and machine learning classifiers.

Electroencephalogram (EEG) data were obtained from 20 subjects performing 54 trials, categorized into rest and fatigue groups. The preprocessing procedures involved artifact elimination, including the removal of noise from muscular activity and eye blinks, as well as bandpass filtering into five frequency bands: delta, theta, alpha, beta, and gamma. Functional connectivity was assessed with the Phase Lag Index (PLI), a reliable metric of phase synchronization among EEG channels, producing high-dimensional datasets. To address the problem of dimensionality, eleven feature selection techniques, such as LASSO, ReliefF, Recursive Feature Elimination with Correlation Bias Reduction (RFE-CBR), and Fisher Score, were employed to discern the features that are most important while preserving interpretability.

Five classifiers were trained using the chosen features: k-Nearest Neighbors (KNN), Support Vector Machine (SVM) with radial basis function (RBF) and linear kernels, Linear Discriminant Analysis (LDA), and Random Forest (RF). Cross-validation methods were employed to guarantee the generalization of the chosen features across different subjects. Performance was assessed utilizing criteria including accuracy, sensitivity, specificity, and F1-Score. The findings indicated that feature selection significantly enhanced classification performance. LASSO was identified as the most effective feature selection algorithm, achieving a combined accuracy of 97.5% with only 19 features and exhibiting excellent performance metrics (accuracy, sensitivity, specificity, and F1-Score) across all classifiers, demonstrating its efficacy in EEG-based fatigue detection.

Lasso identified features that distinguish between rest and fatigue states using EEG channel connections. Connectivity was predominantly focused in the Frontal and Central lobes, indicating their functions in cognitive control and sensorimotor integration. Delta and Theta rhythms exhibited the highest differentiation, indicating their involvement in restorative processes and sustained attention under fatigue. These findings highlight LASSO's accuracy in identifying features relevant to fatigue identification.

The present research illustrates that employing feature selection methods not only reduces the dimensionality of EEG data but also improves model interpretability by concentrating on the most prominent features. The suggested methodology establishes a basis for additional studies in EEG-based fatigue detection and presents possible applications in clinical and working environments, where the assessment of mental fatigue is essential for enhancing safety and performance.

Keywords: Mental Fatigue, EEG (Electroencephalography), Feature Selection, Phase Lag Index (PLI), Machine Learning, Functional Connectivity, Brain Networks

Summary

Mental fatigue, a common condition that negatively impacts cognitive processes, decision-making, and productivity, presents considerable hazards in fields requiring continuous attention and efficiency, including healthcare, transportation, and military operations. The increasing complexity in modern tasks and the dependence on human cognitive capacities underscore the importance of reducing mental fatigue. Understanding and mitigating this condition is crucial for improving safety and operational efficiency in critical conditions. Recent developments in neuroimaging and machine learning offer remarkable possibilities to create accurate and scalable methods for identifying and addressing fatigue. Electroencephalography (EEG), a non-invasive technique for recording cerebral activity, represents a fundamental tool of this research. EEG provides insights into the neural activities linked to cognitive states by monitoring the brain's electrical signals.

This thesis utilizes advanced feature selection algorithms and machine learning classifiers on EEG signals to establish an effective framework for categorizing fatigue states. The incorporation of sophisticated analytical methods improves the interpretability and efficiency of EEG-based evaluations, making them appropriate for practical use. The research included 20 subjects performing 54 trials, whose EEG data were carefully analyzed to identify important connectivity features. The study identifies LASSO regression as the most successful feature selection approach, with an accuracy of 97.5% with a minimal feature set through the application of feature selection techniques and classification algorithms. This research highlights the transformative value of machine learning in developing fatigue detection systems across clinical, technical, and operational environments.

Chapter 1 establishes the theoretical framework of brain networks in order to clarify the neurological basis of mental fatigue. The human brain, comprising approximately 86 billion neurons and trillions of synapses, functions as a highly integrated and dynamic system. In contrast to classical neuroscience, which frequently focused on discrete brain areas, modern methodologies underscore the brain's integrative characteristics via network science. This framework explains how related regions cooperate to facilitate cognition and adjust for different demands. Neural networks display multiple distinct characteristics. Modularity denotes the brain's arrangement into clusters, or modules, that handle particular categories of information. For example, sensory processing and executive functions are regulated by separate yet interconnected areas. This modular design facilitates effective and focused information processing while reducing interference from unrelated regions. Modularity enables functional specialization, permitting the brain to perform complex functions via parallel processing streams.

Small-world architecture is an important characteristic of brain networks, facilitating local processing inside modules and global integration throughout the entire network. This framework facilitates swift communication across remote areas via rather brief routes, guaranteeing both speed and reliability. A small-world network may transmit sensory inputs to decision-making regions with little latency, an essential need in situations demanding rapid reflexes and accurate evaluations. Hubs are crucial for brain connectivity. These interconnected nodes function as communication hubs that consolidate information from multiple modules. The prefrontal cortex, a hub linked to advanced cognitive abilities, illustrates this integrative role. Disruptions to hub connectivity, as seen during mental fatigue, might hinder the brain's capacity to coordinate complex tasks, resulting in diminished efficiency and increased error rates.

Neural circuits associated with sensory perception integrate into broader networks that regulate attention and decision-making. Mental fatigue disrupts this hierarchy, especially in the frontal and central regions, resulting in diminished cognitive

ability and prolonged response times. Investigating brain networks offers a foundation for understanding mental states and drives the development of EEG-based fatigue monitoring systems. By focusing on particular connection patterns, such as phase synchronization, these systems can attain improved accuracy and reliability.

In **Chapter 2**, the materials and methods employed in this study are thoroughly detailed. This study employs a multi-stage methodology to acquire, preprocess, and analyze EEG data, providing a comprehensive framework for identifying mental fatigue. 20 Participants aged 22 to 40 were recruited from medical facilities and divided into two groups: a control group in a rested condition and a fatigue group experiencing sleep loss. EEG recordings were performed from 54 trials using a 64-channel Biosemi ActiveTwo system, conforming to the internationally accepted 10-20 electrode placement standard, hence ensuring consistency and comparability in electrode positioning. EEG signals, inherently susceptible to noise and artifacts, were preprocessed in order to improve data quality. Artifacts, including muscular activity and eye blinks, were systematically removed through advanced signal processing techniques. The data were further categorized into five standard frequency bands: delta, theta, alpha, beta, and gamma. Each band represents specific neurophysiological states, from deep relaxation to increased cognitive activity.

Functional connectivity was evaluated using the Phase Lag Index (PLI), a reliable metric of phase synchronization that reduces the confounding influences of volume conduction. PLI matrices were calculated for each participant, reflecting the complex dynamics of brain connections across frequency bands. These matrices denote high-dimensional datasets that illustrate the brain's functional connection across various cognitive states. Dimensionality reduction was accomplished using eleven sophisticated feature selection methods, each selected for its distinct advantages in managing high-dimensional data. LASSO regression, recognized for its ability to enforce sparsity, proved to be the most efficient technique, reducing the dataset to a minimal subset of characteristics while maintaining interpretability. Alternative techniques, such as ReliefF and Recursive Feature Elimination with Correlation Bias Reduction (RFE-CBR), extended this methodology by prioritizing features that maximize classification performance while minimizing redundancy. Five machine learning classifiers were trained using the selected features: k-Nearest Neighbors (kNN), Support Vector Machines (SVM) with linear and radial basis function (RBF) kernels, Linear Discriminant Analysis (LDA), and Random Forest (RF). Each classifier underwent detailed assessment employing Leave-One-Subject-Out (LOSO) cross-validation, a technique that guarantees strong generalizability among different individuals. Performance indicators, such as accuracy, sensitivity, specificity, and F1-score, were calculated to evaluate the classifiers' performance.

Chapter 3 systematically examines the findings of this thesis, illustrating the reliability of the proposed methodology in effectively differentiating between rested and fatigued states. LASSO regression proved to be the most efficient feature selection strategy, attaining a classification accuracy of 97.5% with merely 19 features. These characteristics were predominantly linked to connection in the frontal and central brain regions, underscoring their essential function in cognitive control and fatigue identification. This outcome highlights the capacity to compress high-dimensional EEG data into a limited feature set without compromising performance. Moreover, all classifiers exhibited significant enhancements in performance when their hyperparameters were optimized with Bayesian Optimization. The SVM utilizing an RBF kernel and Random Forest frequently attained accuracy rates beyond 95%, accompanied by nearly perfect sensitivity and specificity measurements. The results underscore the excellent combination of sophisticated feature selection methods and robust machine learning classifiers in producing dependable classification results.

The connectivity analysis revealed significant differences in delta and theta frequency bands between rested and fatigued states. Delta rhythms, associated with restorative functions, and theta rhythms, related to sustained attention, demonstrated

less synchronization in fatigued conditions. These data correlate with theoretical models of cognitive load and reduced neural efficiency during fatigue, providing a comprehensive view of the underlying brain dynamics. A noteworthy finding is the interpretability of the reduced feature set. The selected features correspond to specific EEG channel connections in the frontal and central lobes, providing insights into the neural mechanisms underlying fatigue. This level of interpretability improves the translational potential of the findings, facilitating practical implementations in real-time fatigue monitoring systems.

Finally, **chapters 4 and 5** summarize the main contributions and conclusions of the proposed fatigue detection framework, address the limitations and lists some ideas for future extensions and further advancements.

This thesis introduces a robust and scalable framework for detecting mental fatigue through EEG, incorporating advanced feature selection and machine learning methodologies. By emphasizing the critical role of frontal and central brain connectivity and the differential significance of delta and theta rhythms, the study delineates vital neural markers of fatigue. Future studies could extend these results by utilizing larger, more heterogeneous datasets and examining practical applications in operational scenarios. The combination of EEG technology and machine learning indicates a frontier with substantial promise for enhancing cognitive health monitoring and improving safety in critical sectors such as healthcare, transportation, and occupational safety.

Acknowledgments

All those who have provided me with support and guidance during the completion of this thesis deserves my most profound appreciation.

First and foremost, I would like to express my sincere gratitude to my professor, G. Matsopoulos, for his invaluable guidance, wisdom, and encouragement. The direction and quality of this work have been significantly influenced by his insightful feedback and expertise.

I am profoundly grateful to Kakkos Ioannis, researcher at the Biomedical Engineering Laboratory of NTUA, for his unwavering support, patience, and dedication. His mentorship has been an inspiration and has helped me navigate through challenges with confidence and clarity.

Finally, I would like to express my gratitude to my wife, Eirini for her constant encouragement and understanding during this endeavor. Her confidence in me has been my most significant source of inspiration and fortitude.

Table of Contents

Abstract.....	5
Summary.....	7
Acknowledgments	11
Table of Contents.....	13
List Of Tables.....	14
List of Figures.....	15
1 Brain Networks.....	16
1.1 Monitoring the Electrical Activity of the Human Brain.....	16
1.2 EEG Electrode Placement.....	16
1.3 EEG Rhythms and Oscillations.....	17
1.4 Introduction to Brain Networks.....	18
1.5 Applications of Network Theory in Neuroscience.....	19
1.6 EEG and ML in Cognitive Workload Assessment.....	20
2 Materials and Methods.....	22
2.1 Data Acquisition Protocol and Recruitment Criteria.....	22
2.2 EEG Data Analysis Pipeline.....	23
2.2.1 EEG Preprocessing.....	24
2.2.2 Feature Extraction Using Phase Lag Index (PLI).....	25
2.2.3 Feature Selection Algorithms.....	26
2.2.4 Classification Algorithms.....	34
3 EEG Analysis Results.....	40
3.1 Classification Results.....	40
3.2 Hyperparameters Optimization Results.....	53
3.3 Performance Metrics.....	65
3.4 Functional Connectivity Features.....	70
3.4.1 Brain Wave band analysis.....	71
3.4.2 Channel Connection Analysis.....	73
3.4.3 Topological Connectivity Analysis.....	76
4 Discussion.....	78
5 Conclusion and Future Considerations.....	80
Bibliographical References.....	82
Appendix.....	88
Matlab Code.....	88
Main Program.....	88
Feature Selection.....	91
Classification.....	93
Classification Optimization.....	95
Plotting Results and EEG Connections.....	97

List Of Tables

Table 1: Classifiers Optimization	53
Table 2: ILFS Results.....	66
Table 3: ReliefF Results	66
Table 4: MutLnfFS Results	66
Table 5: FSV Results	66
Table 6: Laplacian Score Results	67
Table 7: Fisher Score Results	67
Table 8: LLCFS Results	67
Table 9: CFS Results	67
Table 10: LASSO Results	68
Table 11: RFB-CBR-Linear Results.....	68
Table 12: RFB-CBR-Gaussian Results.....	68
Table 13: Most Significant Features Identified By LASSO	70
Table 14: Distribution of connections in frequency bands.....	71
Table 15: Topology of connections.....	73
Table 16: Topology of brain wave connections.....	77

List of Figures

Figure 1: Electrode locations of 64 EEG channel according to the International 10-20 system.....	17
Figure 2: Brain wave frequency bands for EEG signal analysis.....	18
Figure 3: EEG Data analysis Pipeline.....	23
Figure 4: ILFS Accuracy of 5 classifiers.....	41
Figure 5: ILFS Product of 5 classifiers accuracy.....	41
Figure 6: ReliefF Accuracy of 5 classifiers.....	42
Figure 7: ReliefF Product of 5 classifiers accuracy.....	42
Figure 8: Mutinffs Accuracy of 5 classifiers.....	43
Figure 9: Mutinffs Product of 5 classifiers accuracy.....	43
Figure 10: FSV Accuracy of 5 classifiers.....	44
Figure 11: FSV Product of 5 classifiers accuracy.....	44
Figure 12: Laplacian Score Accuracy of 5 classifiers.....	45
Figure 13: Laplacian Score Product of 5 classifiers accuracy.....	45
Figure 14: Fisher Score Accuracy of 5 classifiers.....	46
Figure 15: Fisher Score Product of 5 classifiers accuracy.....	46
Figure 16: LLCFS Accuracy of 5 classifiers.....	47
Figure 17: LLCFS Product of 5 classifiers accuracy.....	47
Figure 18: CFS Accuracy of 5 classifiers.....	48
Figure 19: CFS Product of 5 classifiers accuracy.....	48
Figure 20: Lasso Accuracy of 5 classifiers.....	49
Figure 21: Lasso Product of 5 classifiers accuracy.....	49
Figure 22: RFC-CBR-linear Accuracy of 5 classifiers.....	50
Figure 23: RFC-CBR-linear Product of 5 classifiers accuracy.....	50
Figure 24: RFC-CBR-gaussian Accuracy of 5 classifiers.....	51
Figure 25: RFC-CBR-gaussian Product of 5 classifiers accuracy.....	51
Figure 26: ILFS Accuracy of 5 Optimized classifiers.....	54
Figure 27: ILFS Product of 5 Optimized classifiers accuracy.....	54
Figure 28: ReliefF Accuracy of 5 Optimized classifiers.....	55
Figure 29: ReliefF Product of 5 Optimized classifiers accuracy.....	55
Figure 30: Mutinffs Accuracy of 5 Optimized classifiers.....	56
Figure 31: MutInffFS Product of 5 Optimized classifiers accuracy.....	56
Figure 32: FSV Accuracy of 5 Optimized classifiers.....	57
Figure 33: FSV Product of 5 Optimized classifiers accuracy.....	57
Figure 34: Laplacian Score Accuracy of 5 Optimized classifiers.....	58
Figure 35: Laplacian Score Product of 5 Optimized classifiers accuracy.....	58
Figure 36: Fisher Score Accuracy of 5 Optimized classifiers.....	59
Figure 37: Fisher Score Product of 5 Optimized classifiers accuracy.....	59
Figure 38: LLCFS Accuracy of 5 Optimized classifiers.....	60
Figure 39: LLCFS Product of 5 Optimized classifiers accuracy.....	60
Figure 40: CFS Accuracy of 5 Optimized classifiers.....	61
Figure 41: LLCFS Product of 5 Optimized classifiers accuracy.....	61
Figure 42: LASSO Accuracy of 5 Optimized classifiers.....	62
Figure 43: LASSO Product of 5 Optimized classifiers accuracy.....	62
Figure 44: RFC-CBR-linear Accuracy of 5 Optimized classifiers.....	63
Figure 45: RFC-CBR-linear Product of 5 Optimized classifiers accuracy.....	63
Figure 46: RFC-CBR-gaussian Accuracy of 5 Optimized classifiers.....	64
Figure 47: RFC-CBR-gaussian Product of 5 Optimized classifiers accuracy.....	64
Figure 48: Frequency Band Connection Distribution.....	71
Figure 49: Electrode locations of 64 EEG channel.....	73
Figure 50: Distribution of EEG channel connections by Brain region.....	74
Figure 51: Brain wave connections across different brain lobes.....	76

1 Brain Networks

1.1 Monitoring the Electrical Activity of the Human Brain

The human brain, the central organ of the nervous system, currently consists of numerous unique structures, each of which is responsible for the processing, integration, and coordination of received information. The cerebrum, medulla, and cerebellum are the brain's fundamental components. The cerebrum is further divided into two hemispheres, which are connected by a network of nerve cells known as neurons. Communication is predominantly facilitated by electrical impulses, which are transmitted through the approximately 86 billion neurons in the human brain. The release of neurotransmitters is induced by these impulses, resulting in the formation of sophisticated neural pathways and circuits. The synchronized activity of large groups of neurons is represented by the subtle electrical fields generated during neural activity, which are referred to as post-synaptic potentials. The brain's electrical activity is significantly reflected in these electrical signals when they are averaged across thousands of neurons. This activity can be monitored as it propagates through various anatomical layers, such as the cranium, skin, hair, and meninges, using electrodes inserted on the scalp's surface.[1], [2]

The electroencephalogram (EEG) is one of the most frequently employed techniques for recording brain activity. EEG is a non-invasive method that quantifies voltage fluctuations that are induced by ion flow within oriented neuron populations located near the cerebrum's surface. EEG provides numerous advantages over other brain imaging methods, such as exceptional temporal resolution, which enables the capture of thousands of snapshots of electrical activity from multiple electrodes per second. Furthermore, EEG is cost-effective, and the recordings are passive, which means they do not disrupt brain activity. Electroencephalography (EEG) devices utilize electrodes to record brain wave patterns and transmit the data to a computer for analysis.

There are two categories of EEG electrodes: moist and dry. Wet electrodes are small containers composed of materials such as tin, gold, silver, or stainless steel that are coated with silver chloride. An electrolytic gel serves as a conductor between the cranium and the electrodes [3]. On the contrary, bare electrodes are entirely dependent on mechanical contact [4]. They are substantially smaller than moist electrodes and do not necessitate skin preparation or electrolyte gel. Nevertheless, dry EEG systems currently support a smaller number of channels than moist EEG systems, with a typical limit of up to 30 channels. The primary focus of this thesis will be on high-density moist electrode EEG systems with 32–64 channels. Despite the fact that dry EEG systems are more practicable for real-world applications, such as wearable EEG devices, high-density moist electrode systems offer the necessary spatial resolution for detailed research and analysis.

1.2 EEG Electrode Placement

In order to achieve a standardized interpretation of EEG recordings, internationally recognized methods for describing and applying the placement of scalp electrodes have been established. The most commonly used systems are the “10-10” and, more frequently, the “10–20” systems, which relate the positions of electrodes to the underlying brain regions. In these systems, the terms “10” and “20” refer to the distances between adjacent electrodes, which are either 10% or 20% of the total front-to-back or right-to-left distance of the skull [2]. The experiments conducted in this Thesis (Chapter 2.1) utilize the 10–20 system, where each electrode placement corresponds to a specific brain lobe and area. In this system, even-numbered

electrodes are located on the right side of the scalp, while odd-numbered electrodes correspond to the left side (Figure 1). Additionally, electrodes marked with a "z" indicate midline sagittal positions (e.g., Fpz, Fz, Cz, Oz).

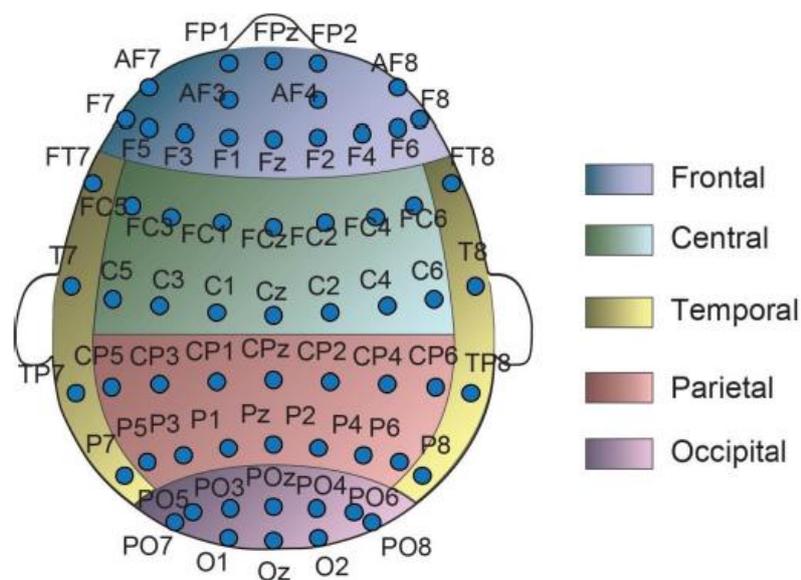


Figure 1: Electrode locations of 64 EEG channel according to the International 10-20 system.

The system identifies the main recording areas as pre-frontal (Fp), frontal (F), temporal (T), parietal (P), occipital (O), and central (C). Furthermore, intermediate electrode placements are labeled using combinations of these designations: AF (between Fp and F), FC (between F and C), FT (between F and T), CP (between C and P), TP (between T and P), and PO (between P and O). "M" electrodes are typically used to mark mastoid areas, located just behind the outer ear, while the "Iz" electrode is placed over the inion. Although mastoid and inion positions are commonly recorded, these locations are often used as fiducial points and do not generally represent higher-order cognitive processes [5].

1.3 EEG Rhythms and Oscillations

EEG signals can be characterized by their rhythmic activity, which is based on the signal morphologies of specific oscillations, defined by the frequencies of their harmonic components (spectral components). These oscillations are categorized into frequency bandwidths that are associated with brain function and condition. The five primary sub-bands are delta (δ , 0.5–3.5 Hz), theta (θ , 3.5–7 Hz), alpha (α , 7–15 Hz), beta (β , 15–30 Hz), and gamma (γ , 30–70 Hz) [6] (Figure 2). It is important to note that the precise boundaries of these sub-bands are not universally defined, resulting in slight variations between studies. Additionally, while some research explores lower or higher frequency ranges, activity outside these bands is often considered artifactual under standard clinical recording techniques.

In addition to frequency bands, other spectral components include power characteristics, which represent the amount of energy in a specific frequency band (commonly expressed as squared amplitude), and phase characteristics, which describe the synchronization of activity across multiple neural generators. Numerous theories have been proposed to explain how factors such as illness, age, and external

stimuli influence changes in internal amplitude and synchronization patterns [7], [8].

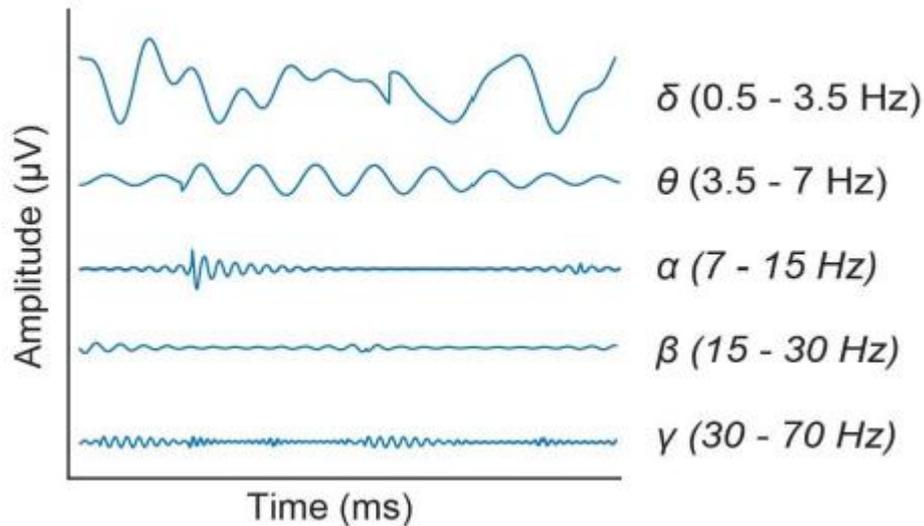


Figure 2: Brain wave frequency bands for EEG signal analysis

1.4 Introduction to Brain Networks

The brain operates as an extensive and linked network, with billions of neurons and trillions of synapses that provide organized routes across various sizes. This complex system facilitates all cognitive, sensory, and motor activities. Conventional research frequently concentrated on solitary neurons or certain brain areas in isolation. Nonetheless, comprehending the brain's genuine activity necessitates analyzing the interactions among these pieces and their formation of coherent networks. These connections, commonly referred to as brain networks, govern behaviors, cognition, and perception. A network-based approach to neuroscience provides insights inaccessible through standalone investigations, as it incorporates the brain's integrative and distributed characteristics[9], [10].

Network science, a field dedicated to comprehending complex systems by the analysis of connected structures, has emerged as a potent instrument in neuroscience. This methodology describes the brain as a network of nodes and edges, with nodes symbolizing either individual neurons or larger cerebral areas, and edges denoting the connections or paths among them. The application of network science in this context allows neuroscientists to measure and evaluate the arrangement of connections, therefore revealing fundamental principles of brain structure and function. This approach enables researchers to progress from analyzing individual components to comprehending the communication and mutual support across brain regions within a cohesive system [11], [12].

Brain networks are characterized by many key properties: modularity, small-world architecture, and the presence of hubs. Modularity is a fundamental characteristic, denoting the brain's structure into discrete clusters, or modules, wherein areas with comparable functions have extensive interconnections within their group. This modular structure allows regions to execute specific functions well, while minimizing interference from other sectors. Each module can perform distinct cognitive or sensory activities while maintaining connectivity with other modules, promoting a balance between specialization and integration. Visual and motor activities may depend on separate modules, however they are interconnected to provide coordinated responses to visual inputs [9], [11], [13], [14].

Another crucial characteristic is the brain's small-world architecture, a configuration that integrates strong local clustering with short pathways linking distant areas. In a small-world network, each node exhibits a high degree of

connectivity to its surrounding nodes, creating densely packed local circuits, while long-range connections interlink these clusters across the network. This design addresses two essential requirements of cerebral function: efficient local processing and effective global integration. The brain's small-world architecture facilitates fast information transmission across distant areas, guaranteeing swift reactions to internal and external stimuli while promoting flexible and robust cognitive functions [15], [16], [17], [18].

Hub regions reflect a third distinctive property of brain networks. These hubs are interconnected nodes that function as essential sites for communication throughout the network. They connect several modules and promote the transmission of information throughout the brain, facilitating coherent and coordinated neural activity. Hub areas are essential to the brain's structural and functional integrity, since they prevent localized processing from occurring in isolation from other modules. Disruptions to these hub locations might yield extensive repercussions, as they impact many modules and obstruct inter-regional communication. Neurological diseases demonstrate that injury to certain hubs leads to considerable cognitive and behavioral disabilities [9], [10], [12], [14], [19], [20], [21].

Network theory's application to neuroscience has yielded significant insights into brain organization, clarifying the relationship between connection patterns and cognitive processes including attention, memory, and decision-making. Researchers can simulate how distinct connection patterns, through concepts like as modularity, small-world organization, and hub connectivity, contribute to functional capacity, adaptation, and resilience. This concept provides a foundation for comprehending network disruptions seen in neurological illnesses, where connection deficits may result in cognitive deterioration and behavioral alterations [10], [11], [22], [23].

1.5 Applications of Network Theory in Neuroscience

The field of neuroscience has been significantly broadened by network theory, which has provided transformative insights into brain structure, function, and pathology. Network-based analysis in neuroscience allows researchers to examine the brain as an interconnected system, rather than as isolated regions. This revelation reveals the intricate interactions that underlie cognition, behaviour, and neurological disorders. This method enables neuroscientists to address concerns in the clinical, developmental, and cognitive domains by emphasizing the connections and influence within brain networks.

Clinical neuroscience is one of the most significant applications of network theory. Network analysis is being employed by researchers to identify disruptions in brain connectivity that are indicative of neurological and psychiatric disorders. For example, distinct patterns of disrupted connectivity within structural and functional brain networks are observed in conditions such as Alzheimer's disease, schizophrenia, and autism. Alzheimer's disease is linked to the disintegration of long-distance connections that connect brain regions, resulting in cognitive impairments such as memory loss and general cognitive decline. Conversely, schizophrenia frequently manifests anomalous connectivity patterns in core regions that are crucial for the integration of cognitive and emotional processes, which contribute to symptoms such as hallucinations and impaired executive functioning [9], [14], [18], [24], [25].

In addition, network measures that are derived from brain imaging can function as biomarkers for the early diagnosis and therapy monitoring. Using imaging technologies such as diffusion tensor imaging (DTI) and functional MRI (fMRI), researchers can identify network abnormalities associated with neurological disorders prior to the emergence of clinical symptoms. This is achieved by evaluating their structural and functional connectivity patterns. In conditions that are progressive, such as Alzheimer's, this ability to detect symptoms early is particularly advantageous, as early intervention may potentially delay or mitigate the onset of symptoms. In addition, network analysis enables clinicians to monitor connectivity patterns over time, thereby assessing the efficacy of treatments as network integrity improves. [13], [14], [19], [24].

Pharmacology also benefits from network theory, as it clarifies the manner in which pharmaceuticals influence brain connectivity. Dopamine-modulating medications, for instance, have demonstrated potential treatments for conditions such as depression and Parkinson's disease by enhancing the efficiency of functional networks. In disorders where neurotransmitter imbalances disrupt typical connectivity patterns, therapeutic pathways for restoring network function can be provided by understanding how drugs influence network connectivity [13], [14], [22], [25].

Network analysis has improved our comprehension of cognitive functions that arise from distributed brain systems in cognitive neuroscience. Dynamic interactions across networks are the foundation of processes such as executive function, attention, and memory, rather than dependence on individual regions. The brain's modular structure enables specialized processing, while its small-world and hierarchical features facilitate the rapid transmission of information across regions. For example, the default mode network (DMN) is active during repose and is involved in self-referential thought, memory consolidation, and planning. The DMN's connectivity abnormalities have been associated with mental health conditions, highlighting the role of network dynamics in the development of cognitive health and disease [13], [25].

Network theory is also beneficial to developmental neuroscience, particularly in the context of investigating changes in brain connectivity throughout the lifespan. Modular integration and long-range connectivity are particularly important during adolescence, a time when individuals are more susceptible to mental health issues. These observations have the potential to inform translational applications, in which network-focused interventions such as cognitive training or neurostimulation can enhance healthy connectivity patterns during critical developmental stages [22], [24].

In neuroscience, network theory has become an essential instrument, enhancing our comprehension of cognitive functions and advancing methods for diagnosing, monitoring, and treating neurological conditions. Network theory provides the way for future research and applications in clinical and experimental neuroscience by emphasizing the brain's inherent connectivity and integrative properties.

1.6 EEG and ML in Cognitive Workload Assessment

Particularly during demanding activities, mental tasks comprise a diverse array of cognitive operations, including information storage, processing, transfer, and retrieval, which can impede performance [26]. A reallocation of cognitive resources is necessary for sustained high-efficiency mental efforts, particularly in challenging tasks, which ultimately increases the mental workload [27]. Recent research suggests that increased work engagement or intensity can result in mental fatigue, reduced operational performance, and health issues such as exhaustion syndrome [28], [29]. The potential for real-world evaluation of cognitive burdens is presented by the accurate assessment of workload-related mental states, which has the potential to be applied in clinical settings [30], [31]. In spite of the extensive research conducted on cognitive workload [28], the comprehension of the brain's functions and the mechanisms that regulate them remains a challenge, particularly in the context of real-world EEG workload detection, where methodological constraints limit practical applications.

The electroencephalogram (EEG) is a non-invasive, cost-effective, and practical method for measuring brain activity with high temporal resolution [32]. Studies have shown that EEG brain oscillations are strongly correlated with cognitive burden, identifying neuronal processes that serve as reliable indicators of workload [33], [34]. It has been demonstrated that brain wave patterns, including parieto-occipital alpha and temporal beta power [35], as well as alpha/theta power in frontal and posterior regions [36], [37], are influenced by task complexity and exertion.

The effective monitoring and interpretation of mental stress mechanisms are facilitated by machine learning (ML) techniques, which are necessary due to the complexity of neural oscillatory activity across various brain regions. Despite the fact

that the majority of ML studies concentrate on classification performance without thoroughly considering physiological noise or data quality, the capacity to differentiate cognitive states using EEG features supports the development of high-performing, transparent classification models. The integration of ML with EEG-based features not only improves the discrimination between workload-related cognitive states but also reveals underlying mental processes associated with brain wave frequencies and cranium locations [38], [39], [40], [41]. For instance, Wang et al. [41] utilized a proximal support vector machine (SVM) in a four-level working memory task to achieve superior performance. They capitalized on EEG features such as alpha, frontal theta, and posterior high beta and low gamma bands. Similarly, research has emphasized the importance of delta band activity in the evaluation of workload, particularly in frontal regions, as well as reductions in brain wave amplitude (e.g., centroparietal alpha and midline beta) [40].

Nevertheless, the challenge of generalizing EEG features across a variety of tasks remains, despite these advancements. Cross-task workload classification frequently experiences diminished accuracy as a result of task-dependent characteristics that fail to capture global workload traits, despite the fact that single-task workload classification has demonstrated reliable performance. Therefore, the classification accuracy in cross-task scenarios frequently remains at or near chance levels [42], [43], [44], with only a handful of exceptions attaining dependable results [45]. In order to enhance performance, research has investigated the integration of a variety of cognitive attributes, including spatial, spectral, and temporal EEG features, or the integration of features from other electrophysiological modalities [46], [47]. The accuracy of cross-task classification can be improved by the introduction of new features [47], [48].

Recent research has demonstrated that the human connectome is a large-scale network of interconnected regions that the brain operates as. This structure enables the integration of information, functional processing, and neural communication [49]. Insights into these intricate brain functions are provided by functional connectivity (FC) under varying burden conditions, which reveals localized and global processing and modulations of the brain's topological properties as a result of task load effects [49], [50].

The incorporation of FC features into ML frameworks improves interpretability by revealing the brain's functional reorganization and the concealed layers of cognitive processes associated with mental fatigue. This method has demonstrated potential for enhancing classification performance [50], [51], [52]. Nevertheless, there is a scarcity of research that has integrated FC into ML models to examine task-related brain function alterations or brain region communication under varying workload levels in simulated environments [53], [54]. A notable research [55] employed Feature selection features to discriminate between cross-task workloads, resulting in a task-independent classification accuracy of 0.87 through the analysis of frontal theta and beta band power variations.

2 Materials and Methods

2.1 Data Acquisition Protocol and Recruitment Criteria

The data acquisition protocol was structured to investigate the impact of sleep deprivation on brain connectivity and cognitive performance in a clinical setting. Participants were recruited from doctors and nursing staff at the 401 General Military Hospital of Athens, with EEG data collection carried out in collaboration with the Biomedical Optics and Applied Biophysics Laboratory at the National Technical University of Athens. The study included 20 volunteers aged 22 to 40, who met specific inclusion criteria: participants had to be in good general health, free from neurological or psychiatric conditions, and not currently using medications that could affect cognitive performance or brain activity. They were divided into two groups: a fatigue group, consisting of participants who had completed a long shift and were sleep-deprived, and a control group in a rested state. This design allowed for direct comparison between fatigued and non-fatigued participants under identical conditions.

The protocol included a series of cognitive tasks structured to assess mental fatigue and cognitive performance. First, participants completed a resting-state recording by focusing on a fixed point for five minutes to establish baseline neural activity. Following this, they performed the Psychomotor Vigilance Test (PVT), a task designed to measure sustained attention and response speed by requiring participants to react to visual stimuli appearing on a screen at random intervals. The PVT is highly sensitive to sleep deprivation effects and evaluates reaction time consistency.

Two working memory tasks were also included to evaluate cognitive function under conditions of rest and fatigue. The N-Back Task required participants to view images on a screen and identify whether each image matched the one shown two images earlier. This task places demands on spatial working memory, requiring participants to continuously update and recognize visual information. The second working memory task, the Paced Auditory Serial Addition Test (PASAT), involved listening to a sequence of numbers and adding each new number to the one that immediately preceded it. The PASAT measures working memory, processing speed, and attention, challenging participants to process and retain auditory information continuously.

EEG data from 54 trials was collected using a 64-channel Biosemi ActiveTwo system in line with the 10-20 electrode placement standard, capturing brain activity across different scalp regions for 1024 samples at 256Hz Sampling rate. EEG signals were synchronized with behavioural responses to provide insight into the neural dynamics associated with reaction times and task accuracy. Data was later filtered to analyse activity across specific frequency bands, enabling the comparison of connectivity patterns between rest and fatigue conditions across the different cognitive tasks. This protocol allowed for a detailed analysis of how sleep deprivation and mental fatigue influence both cognitive performance and neural connectivity.

Ethical approval for this study was obtained from the institutional review board, ensuring that all procedures complied with ethical standards for research involving human participants. Each participant provided informed consent before participating, fully understanding the study's objectives, procedures, and any potential risks involved. Confidentiality was maintained by anonymizing data, and participants were informed of their right to withdraw from the study at any time without consequence. Special care was taken to ensure the physical and psychological well-being of all participants, especially those undergoing tasks after extended work shifts, to prevent undue stress or discomfort. This protocol adheres to the principles outlined in the Declaration of Helsinki, safeguarding participant rights and welfare throughout the study.

2.2 EEG Data Analysis Pipeline

We implemented using MATLAB® and EEGLAB [56] a detailed pipeline for analyzing EEG data to classify resting and fatigue states. This pipeline is organized into distinct stages, including preprocessing EEG signals, extracting functional connectivity features, applying advanced feature selection methods, performing machine learning classification, and interpreting the results by identifying significant EEG channels and frequency bands. The ultimate aim is to uncover neural patterns indicative of fatigue, providing both practical applications and physiological insights.

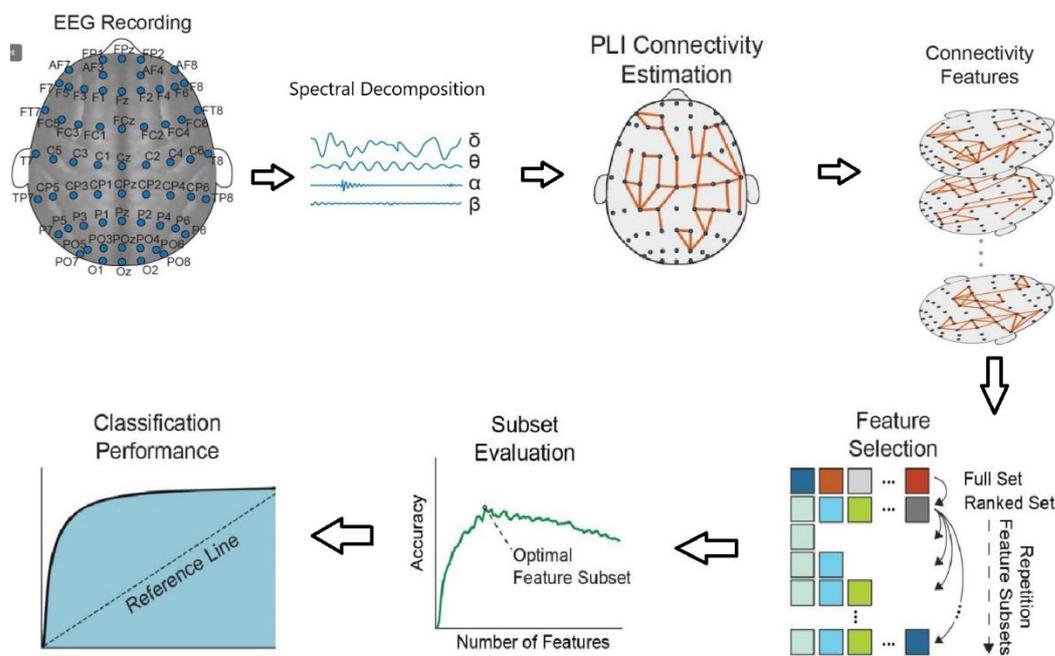


Figure 3: EEG Data analysis Pipeline

The preprocessing stage begins by loading EEG data digitized at a sampling rate of 256 Hz from two conditions: resting and fatigue. Data is organized into batches, with each subject's recordings processed independently. The EEG signals are then filtered into five standard frequency bands—delta (1–4 Hz), theta (4–7 Hz), alpha (8–12 Hz), beta (13–30 Hz), and gamma (31–45 Hz)—using zero-phase finite impulse response (FIR) filters. Each band captures unique neurophysiological states, such as relaxation (theta) or cognitive activity (beta and gamma). After filtering, the Phase Lag Index (PLI) is computed for each frequency band. The PLI measures phase synchronization between EEG channels and reflects functional connectivity in the brain. For each subject and frequency band, PLI matrices are computed and averaged across trials to reduce noise and variability, ensuring robust feature extraction.

The feature extraction step involves processing the PLI matrices to generate feature vectors. Each PLI matrix is symmetrical, representing functional connectivity between pairs of EEG channels. The upper triangular elements, excluding the diagonal, are extracted to form one-dimensional feature vectors. These vectors, representing unique connectivity features, are compiled into a feature matrix where each row corresponds to a subject, and each column represents a connectivity feature. Separate feature matrices are created for the resting and fatigue conditions. These matrices are then merged into a single dataset with binary labels: 0 for resting and 1 for fatigue. This labeled dataset forms the foundation for subsequent feature selection and classification.

To enhance the classification process, feature selection is employed to identify the most relevant features while reducing dimensionality. The pipeline integrates several advanced feature selection algorithms, each offering unique strengths. Infinite Latent Feature Selection (ILFS) identifies features by balancing relevance and

redundancy. ReliefF evaluates features based on their ability to separate instances near decision boundaries, while Mutual Information Feature Selection (MutInfFS) selects features that maximize information gain. Recursive Feature Elimination (RFE) iteratively removes less informative features based on classifier performance, and Correlation-Based Feature Selection (CFS) prioritizes features with high relevance to the class and low inter-correlation. Other methods include the Laplacian Score, which preserves local data geometry, and Fisher Score, which ranks features by class separability. LASSO regression employs L1 regularization to shrink less relevant features to zero, and Local Learning-Based Clustering Feature Selection (LLCFS) selects features that minimize clustering error. Finally, RFE with Correlation Bias Reduction (RFECBR) refines feature selection using linear or Gaussian kernels to reduce bias. These methods are applied using Leave-One-Subject-Out (LOSO) cross-validation to ensure robust and unbiased evaluation. The output is a ranked list of features for each cross-validation fold.

The selected features are used to train and test machine learning classifiers for distinguishing between resting and fatigue states. Five classifiers are implemented, including K-Nearest Neighbors (KNN), which classifies data based on the majority class of the nearest neighbors, and Support Vector Machines (SVM) with both linear and Radial Basis Function (RBF) kernels. Linear Discriminant Analysis (LDA) projects the data onto a lower-dimensional space that maximizes class separability, while Random Forest (RF) employs an ensemble of decision trees to enhance accuracy and reduce overfitting. Each classifier is evaluated using metrics such as accuracy, sensitivity, specificity, and the area under the receiver operating characteristic curve (AUC). The number of features is systematically varied to optimize classification performance, with results visualized to show the relationship between the number of features and accuracy.

The results are aggregated and visualized to provide insights into classifier performance and the effectiveness of feature selection methods. The accuracy of each classifier is averaged across all cross-validation folds, and combined accuracy metrics are computed to determine the optimal feature set. This stage highlights the trade-off between the number of features and classification accuracy, enabling the identification of an optimal feature subset.

2.2.1 EEG Preprocessing

The filtering process in EEG preprocessing is essential for isolating neural oscillations within specific frequency bands, each associated with distinct cognitive and physiological states. In the preprocessing pipeline used in this study, filtering was applied to the EEG signals using EEGLAB's filtering function. EEGLAB [56], a MATLAB® toolbox for EEG analysis, provides tools for data preprocessing, artifact removal, and connectivity measurements. The filtering function allows for precise control over the frequency range, enabling the filtering of EEG data within specific bands to focus on relevant neural activity while eliminating unwanted noise. Finite Impulse Response (FIR) band-pass filters were employed in order to isolate the brain frequency bands [57].

The FIR Filter of N order the output sequence is calculated

$$y[n] = \sum_{i=0}^N b_i x[n - i] \quad (1)$$

where N is the filter order, b_i is the impulse response of the filter, $x[n]$ is the input signal, and $y[n]$ is the output signal.

Each frequency band is associated with distinct neural processes and cognitive states, making it crucial to separate them for targeted analysis.

1. **Delta Waves (1-4 Hz):** The lowest frequency band, delta waves are prominent during deep sleep stages and are essential for restorative sleep and brain

recovery. Delta waves are often linked to low cognitive arousal and disengagement from external stimuli. Excessive delta activity while awake can indicate fatigue or impaired cognitive processing, as the brain enters a state that resembles sleep-like disengagement.

2. **Theta Waves (4-7 Hz):** Theta waves are often linked to drowsiness, light sleep, and meditation. Theta oscillations are known to play a role in memory processes, learning, and emotional processing, especially during tasks requiring focused internal attention. In the context of fatigue, elevated theta activity is often observed, reflecting decreased alertness and a more relaxed, less vigilant mental state.
3. **Alpha Waves (8-12 Hz):** Alpha waves are typically observed during wakeful relaxation, particularly with eyes closed. They represent an idle state where the brain is not actively processing sensory input. Alpha activity decreases with mental exertion and focused attention, making it a valuable indicator of relaxation versus cognitive engagement. In states of fatigue, alpha activity can increase as the brain moves toward a more restful state, reflecting a decline in active, alert processing.
4. **Beta Waves (13-30 Hz):** Beta waves are associated with active thinking, problem-solving, and high mental engagement. This frequency band is linked to focused attention, concentration, and heightened cognitive performance. During states of fatigue, beta activity often decreases as the brain's capacity for sustained attention and executive processing diminishes. A drop in beta waves can signal reduced cognitive control and slower response times in demanding tasks.
5. **Gamma Waves (31-45 Hz):** Gamma waves are the highest frequency band typically analysed in EEG and are associated with higher-order cognitive functions, such as perception, attention, and memory integration. Gamma activity is crucial for complex information processing and is involved in multi-sensory integration and conscious perception. Fatigue may disrupt gamma activity, reflecting decreased cognitive efficiency and slower processing speeds in high-demand tasks.

By bandpass filtering the EEG data into these frequency-specific bands, the preprocessing step helps to isolate the oscillatory activity most relevant for understanding brain states related to rest and fatigue. Filtering enables precise analysis by discarding frequencies outside the target range, ensuring that only relevant brain wave information is used in further analyses.

2.2.2 Feature Extraction Using Phase Lag Index (PLI)

The Phase Lag Index (PLI) is a connectivity measure commonly used in EEG analysis to assess the synchronization between different brain regions [11], [13]. Specifically, PLI quantifies the consistency of phase differences between pairs of EEG signals over time, offering insights into the functional connectivity within the brain. It is particularly valuable because it is robust against volume conduction effects, which can distort connectivity estimates by creating artificial correlations between electrodes in proximity. By focusing on phase differences rather than amplitude correlations, PLI provides a more accurate representation of the underlying neural interactions.

PLI computation starts by calculating the instantaneous phase of each EEG signal, often using the Hilbert transform or a similar time-frequency analysis method. The phase of a signal at a given time point reflects the position within its oscillatory cycle (e.g., peak, trough, or mid-cycle). For two signals, the phase difference at each time point can be determined by subtracting one phase from the other. When two signals are in sync, their phase differences remain stable, either consistently positive or negative. However, if the signals are out of sync or vary in synchronization, the phase differences will fluctuate around zero.

The PLI is calculated by determining the consistency in the sign of these phase differences over time. Specifically, it is the proportion of time points where the phase difference between two signals does not equal zero. Mathematically, PLI is defined as:

$$PLI = \left| \frac{1}{N} \sum_{t=1}^N \text{sign}(\Delta \varphi_t) \right| \quad (2)$$

where N is the number of time points and $\Delta \varphi_t$ represents the phase difference at time t . The sign function is applied to the phase difference to obtain only the direction (positive or negative) of phase lag, ignoring the exact magnitude. PLI values range from 0 to 1: a PLI of 0 indicates no consistent phase relationship, while a PLI close to 1 reflects a strong, consistent lag in phase, suggesting a directional influence or causality between signals.

In practice, PLI is calculated across various frequency bands to capture functional connectivity in specific ranges associated with different cognitive and physiological states. For example, lower frequencies (such as delta and theta bands) are typically linked to cognitive states like attention and memory, while higher frequencies (alpha, beta, and gamma bands) are associated with processes such as perception and motor functions. By analyzing PLI in these distinct frequency bands, researchers can obtain a frequency-specific map of connectivity patterns, which provides more detailed insights into brain dynamics.

PLI is particularly suited for studying connectivity in EEG data because it minimizes the influence of spurious correlations caused by volume conduction. Volume conduction can result in misleading correlations between signals simply due to their spatial proximity rather than actual neural interactions. Since PLI only considers phase shifts that reflect genuine interaction and not zero-phase lag, it is relatively immune to these artifacts, making it a more reliable measure of functional connectivity than simple coherence or correlation-based measures.

2.2.3 Feature Selection Algorithms

To improve classification accuracy, the pipeline incorporates feature selection to identify the most significant features while reducing the dataset's dimensionality. Several advanced algorithms are utilized, each offering distinct advantages. Infinite Latent Feature Selection (ILFS) selects features by balancing their relevance and redundancy. ReliefF focuses on features that effectively differentiate instances near decision boundaries, while Mutual Information Feature Selection (MutInfFS) targets features that maximize information gain. Recursive Feature Elimination (RFE) systematically removes the least important features based on classifier performance, and Correlation-Based Feature Selection (CFS) emphasizes features with strong correlations to the target class but low inter-correlation among themselves. Additional techniques include the Laplacian Score, which maintains the local geometric structure of the data, and the Fisher Score, which ranks features based on their ability to distinguish between classes. LASSO regression applies L1 regularization to eliminate less relevant features, and Local Learning-Based Clustering Feature Selection (LLCFS) minimizes clustering errors to identify key features. Lastly, RFE with Correlation Bias Reduction (RFECBR) improves feature selection by addressing correlation bias, employing either linear or Gaussian kernels. These methods are applied within a Leave-One-Subject-Out (LOSO) cross-validation framework, ensuring unbiased and robust evaluation.

2.2.3.1 Infinite Latent Feature Selection (ILFS)

Infinite Latent Feature Selection (ILFS) is a sophisticated feature selection technique intended for high-dimensional datasets characterized by complex architecture. Conventional feature selection techniques frequently focus on statistical metrics that evaluate features in isolation or examine pairwise correlations, which may be inadequate for revealing the complex interdependencies that characterize intricate datasets such as electroencephalography (EEG) signals. ILFS mitigates this disadvantage by creating an infinite feature selection graph that represents both global and local data structures, facilitating the discovery of features that are meaningful, non-redundant, and representational of the underlying data manifold [58].

The core concept of ILFS involves showing the connections between characteristics and samples via a graph-based methodology. ILFS assesses the significance of each feature by examining an unlimited array of potential feature subsets, focusing on its role in reconstructing the complete dataset. This is accomplished by computing a feature relevance score using pseudo-inverse projections, which measure the extent to which one feature may be linearly approximated by others. Features that substantially influence the reconstruction error are considered more significant, since they offer distinct information not represented by other features.

ILFS enhances feature rankings by integrating locality information, so assuring that the chosen features maintain the local neighborhood structure of the data. This is especially significant in datasets where local patterns are essential for differentiating across classes. ILFS adeptly finds characteristics that augment the discriminative capacity of machine learning models by harmonizing global reconstruction capabilities with local structural preservation.

A significant advantage of ILFS is its parameter-free characteristic, which eliminates the necessity for substantial tuning and enhances its adaptability to many data kinds. Its resilience to strongly correlated features and capacity to capture nonlinear interactions without explicit kernel functions render it appropriate for complex biomedical datasets. Research indicates that ILFS can markedly enhance classification accuracy by decreasing dimensionality while preserving critical information, hence improving computing efficiency and model interpretability.

2.2.3.2 ReliefF

ReliefF is a widely used feature selection algorithm that enhances the original Relief method to accommodate multiclass issues and increase resilience to noisy and incomplete data. ReliefF, a filter-based feature selection method, assesses feature quality by measuring their capacity to differentiate between identical occurrences. It functions by allocating weights to features according to their efficacy in distinguishing between instances of the same class (nearest hits) and examples of other classes (nearest misses) [59], [60].

The procedure begins by randomly choosing a selection of cases from the dataset. For each chosen instance, ReliefF determines its nearest neighbors: one from the same class and one from each of the other classes in the dataset. Subsequently, it adjusts the weight of each feature based on the difference between the feature values of the instance and those of its neighbors. If a feature value differs between the instance and a nearby miss (of a different class), the weight of the feature is augmented, signifying its use in class differentiation. If the feature value varies between the instance and a nearest hit of the same class, the weight is diminished, indicating that the feature may lack reliability for classification.

A fundamental advantage of ReliefF is its capacity to capture feature connections and interactions without presuming feature independence. It is responsive to characteristics that are conditionally dependent on other features, proving it useful in detecting combinatorial patterns within the data. Moreover, ReliefF exhibits greater

computational efficiency than wrapper approaches, as it does not need the training of a classifier for every assessed feature subset.

ReliefF has been effectively utilized across several fields, including as biological signal processing, genetics, and text categorization. Researchers can enhance the effectiveness of machine learning classifiers by sorting features according to their weights and selecting the most informative ones.

2.2.3.3 Mutual Information Feature Selection (MutInfFS)

Mutual Information Feature Selection (MutInfFS) is a statistical technique employed to find and choose pertinent features in high-dimensional datasets, especially in supervised learning contexts. It utilizes the principle of mutual information from information theory, which measures the information one random variable contains relating to another. MutInfFS efficiently quantifies the mutual dependency between each feature and the target variable, capturing both linear and nonlinear interactions without supposing any particular data distribution [61], [62].

In MutInfFS, the mutual information for each feature relative to the class labels is calculated separately. This metric indicates the extent to which the knowledge of a feature's value diminishes the uncertainty of the target variable. Features exhibiting elevated mutual information scores tend to be more informative since they offer substantial insights into the behavior of the target variable. The algorithm orders the features according to these scores in descending sequence, facilitating the selection of the highest-ranked features for model training.

The main advantage of MutInfFS is its capacity to identify complicated, nonlinear connections that conventional correlation-based techniques may neglect. This renders it especially appropriate for domains such as biomedical engineering and bioinformatics, where data frequently display complex interrelations. By concentrating on the most informative attributes, it augments the efficacy of machine learning classifiers, resulting in enhanced accuracy and interpretability.

Nonetheless, the computation of mutual information poses specific difficulties. Estimating mutual information from finite samples can be computationally demanding and necessitates meticulous attention to binning strategies or kernel density estimation methods for continuous variables. These estimate approaches may add bias or volatility, hence impacting the dependability of the mutual information scores.

In practice, MutInfFS is valued for its simplicity and efficacy across several domains. It functions as a filtering technique, rendering it computationally more efficient than wrapper techniques that need model training for each assessed feature subset. MutInfFS accelerates the training of machine learning models by diminishing data dimensionality, hence mitigating the curse of dimensionality, which may enhance model generalization and decrease overfitting.

2.2.3.4 Feature Selection with SVM (FSV)

Feature Selection using Support Vector Machines (FSV) is a supervised approach that utilizes Support Vector Machines (SVMs) to discern the most useful features for classification purposes. In high-dimensional datasets, particularly in biological signal processing and bioinformatics, the selection of pertinent features is essential for optimizing model performance, minimizing computing cost, and augmenting interpretability. FSV employs the characteristics of SVMs, namely their capacity to identify optimum hyperplanes that maximally divide classes, to rank features according to their influence on the decision boundary [63], [64].

The basic theory of FSV entails building a Support Vector Machine (SVM) model on the dataset and examining the weights attributed to each feature. In linear SVMs, the magnitudes of the weight vector components directly indicate the significance of the associated features. Features with greater weights impose a more substantial influence on the hyperplane's placement and, thus, on the classification result. FSV

finds the most significant attributes for class distinction by ranking them based on the size of their weights.

FSV may be generalized to nonlinear SVMs by the utilization of kernel functions, including the radial basis function (RBF) kernel. Nonetheless, evaluating feature significance in nonlinear models is more complicated due to the implicit modification of the feature space. Methods like as Recursive Feature Elimination (RFE) are frequently combined with Support Vector Machines (SVMs) to systematically eliminate the least significant features according to model performance, therefore optimizing the feature set to those that improve classification precision.

Feature selection utilizing SVM is an effective technique that incorporates feature selection directly inside the classification model. By employing the weights obtained from SVM training, FSV prioritizes features according to their influence on the decision boundary, encompassing both the significance of individual features and their interconnections. Despite processing difficulties, FSV provides significant insights into feature significance, facilitating the creation of more precise and interpretable machine learning models, especially in domains involving intricate, high-dimensional data such as EEG signal analysis.

2.2.3.5 Laplacian Score

Laplacian Score is an unsupervised feature selection technique that assesses the significance of features by their capacity to maintain the inherent geometric structure of the data. In contrast to supervised approaches that depend on class labels, the Laplacian Score evaluates features by examining the local manifold structure, making it especially appropriate for datasets with little or inconsistent label information. This approach is based on the ideas of manifold learning and spectral graph theory, utilizing the concept that data points in proximity inside a high-dimensional space should remain adjacent in a lower-dimensional representation[65], [66].

The fundamental principle of the Laplacian Score involves the creation of a nearest-neighbor graph to represent the local associations between data points. Each node in the graph signifies a data sample, and edges link nodes considered neighbors according to a selected distance measure, commonly the Euclidean distance. The weight of each edge indicates the similarity between the associated nodes. The Laplacian Score for each feature is calculated by assessing its effectiveness in preserving the locality structure established by the graph. This includes calculating the variance of the feature, weighted by the Laplacian matrix of the graph, which represents the graph's connectedness and weights.

Features that provide low Laplacian Scores are considered more significant since they more effectively maintain the local neighborhood structure of the data. A low Laplacian Score signifies that the feature exhibits analogous values for proximate data points while displaying distinct values for remote sites, consistent with the manifold assumption that adjacent points are inclined to possess comparable characteristics. The approach efficiently decreases dimensionality by picking features with low Laplacian Scores, while preserving the dataset's fundamental structural information.

The Laplacian Score approach offers an important advantage in its capacity to manage nonlinear patterns included in complicated datasets, including those seen in image processing, text mining, and biological signal analysis, such as EEG data. It is frequently utilized with supervised algorithms to equilibrate the retention of data geometry with class differentiation. It functions as an essential instrument in the preprocessing phase, decreasing dimensionality and emphasizing aspects that are structurally pertinent. Integrating Laplacian Score-based feature selection with machine learning classifiers can boost model performance, improve computational efficiency, and provide deeper insights into the underlying data structures.

2.2.3.6 Fisher Score

Feature Selection with Fisher Score is a supervised method extensively used for identifying and ranking important features in classification tasks, particularly within high-dimensional datasets common in fields like biomedical signal processing. The Fisher Score evaluates each feature individually based on its ability to discriminate between different classes. It operates by measuring the ratio of between-class variance to within-class variance for each feature, thereby quantifying how well a feature can separate the data into distinct categories [67], [68].

Mathematically, the Fisher Score for a given feature is calculated by taking the squared difference between the mean feature values of each class and the overall mean, weighted by the class probabilities, and dividing it by the sum of the variances within each class. A higher Fisher Score indicates that the feature has a greater discriminative power, as it shows significant differences between classes while maintaining consistency within classes. By ranking features according to their Fisher Scores, one can select the top-ranked features that contribute most effectively to class separation.

One of the primary advantages of the Fisher Score method is its simplicity and computational efficiency. It does not require iterative optimization or complex algorithms, making it suitable for large datasets where computational resources may be limited. Additionally, since it evaluates each feature independently, it is easily parallelizable, which can further reduce computation time.

In the context of EEG-based mental fatigue detection, Feature Selection with Fisher Score can effectively identify features that reflect significant differences in brain activity between rest and fatigue states. By selecting features with high Fisher Scores, machine learning classifiers can achieve improved accuracy and generalization, as the selected features provide clear separation between the cognitive states.

2.2.3.7 Local Learning-Based Clustering Feature Selection (LLCFS)

Local Learning-Based Clustering Feature Selection (LLCFS) is an unsupervised feature selection method that is specifically designed to manage high-dimensional datasets with intricate structures. LLCFS concentrates on the identification of the most important characteristics for local learning and clustering tasks by encoding the intrinsic manifold structure of the data. The preservation of local neighborhood relationships among data points is the primary focus of LLCFS, which operates without supervision, in contrast to traditional methods that require class identifiers or rely on global data properties [69], [70].

The fundamental concept of LLCFS is the development of a local learning model that approximates the data distribution in close proximity of each data point. It relies on the assumption that data elements that are in close proximity to one another in the high-dimensional space are likely to exhibit similar characteristics. LLCFS attempts to identify features that most accurately depict the local geometry of the data manifold by modelling these local neighborhoods.

LLCFS employs a clustering-based objective function that evaluates the significance of features by assessing their capacity to preserve local similarities. In particular, it determines a locality-preserving score for each feature by assessing the extent to which the feature contributes to the reconstruction of data points from their neighbors. Features that generate reduced reconstruction errors are prioritized due to their superior preservation of the data's local structure.

The capacity of LLCFS to manage nonlinear data distributions and identify meaningful patterns that may not be detectable by global feature selection methods is a significant advantage. This renders it especially well-suited for applications such as biomedical signal processing, image recognition, and text mining, as well as the analysis of electroencephalography (EEG) data. In the context of EEG signals, LLCFS can identify features that capture subtle, localized patterns of brain activity associated with different cognitive states.

LLCFS effectively reduces dimensionality while retaining essential information by concentrating on local structures, thereby improving the performance and efficacy of machine learning algorithms that are applied subsequently. The technique is also computationally efficient, as it does not require the complex optimization procedures that are common in some wrapper and embedded methods.

2.2.3.8 Correlation-based Feature Selection (CFS)

Correlation-based Feature Selection (CFS) is a filter-based feature selection method that evaluates subsets of features based on their predictive ability and the level of redundancy among them. The fundamental principle of CFS is that a good feature subset is one that contains features highly correlated with the target variable (class) while having low intercorrelation among themselves. By focusing on the selection of feature subsets rather than individual features, CFS aims to identify groups of features that collectively contribute to improved model performance[71]. CFS operates by calculating a merit score for each potential feature subset using the formula:

$$Merit_s = \frac{k\bar{r}_{cf}}{\sqrt{k + (k - 1)\bar{r}_{ff}}} \quad (3)$$

- k is the number of features
- \bar{r}_{cf} is the average feature class correlation
- \bar{r}_{ff} is the average feature-feature correlation

This formula balances the relevance of the features to the class label against the redundancy among the features. A higher merit score indicates a feature subset where features are strongly associated with the class but are weakly correlated with each other.

In practice, CFS employs heuristic search strategies such as best-first search, greedy stepwise selection, or genetic algorithms to navigate the space of possible feature subsets efficiently. These strategies help in managing computational complexity, especially in high-dimensional datasets. By evaluating subsets rather than individual features, CFS accounts for feature interactions and synergy, which can be crucial in complex domains like biomedical signal processing.

A notable feature of CFS is its simplicity and computational efficiency in contrast to wrapper approaches, which need training a classifier for each assessed subset. CFS, as a filter technique, operates independently of any particular learning algorithm, hence augmenting its adaptability and facilitating its inclusion into other preprocessing pipelines. It efficiently lowers dimensionality, resulting in accelerated training durations and perhaps enhancing the generalization efficacy of machine learning models by reducing overfitting.

2.2.3.9 Lasso Regression

Lasso Regression, or Least Absolute Shrinkage and Selection Operator, is a significant method in statistical modelling and machine learning for managing high-dimensional datasets. Lasso Regression concurrently performs variable selection and regularization, hence improving the predictive accuracy and interpretability of the resultant model. It accomplishes this by applying an L1 regularization penalty on the absolute values of the regression coefficients, so reducing some of them to precisely zero. This characteristic inevitably chooses a more straightforward model that preserves just the most crucial characteristics, rendering Lasso an effective instrument for feature selection. [72], [73].

Mathematically, Lasso minimizes the objective function:

$$\min_{\beta} \left(\frac{1}{2n} \sum_{i=1}^n (y_i - X_i \beta)^2 + \lambda \sum_{j=1}^p |\beta_j| \right) \quad (4)$$

where n is the number of observations, p is the number of predictors, y_i is the response variable, X_i represents the predictor variables, β_j are the coefficients, and λ is the regularization parameter controlling the strength of the penalty.

The L1 penalty $\lambda \sum_{j=1}^p |\beta_j|$ encourages sparsity in the coefficient vector β leading to the elimination of less important features.

Lasso Regression is very beneficial in the realm of high-dimensional data, such as electroencephalography (EEG) signals. EEG datasets frequently exhibit a substantial number of characteristics in relation to the number of observations, with many of these features potentially being unnecessary or redundant. Utilizing Lasso facilitates dimensionality reduction by identifying a subset of features that most effectively predict the target variable, thereby enhancing computing efficiency and mitigating the danger of overfitting.

Lasso Regression's primary advantage in feature selection is its capacity to manage multicollinearity among predictors. In scenarios where predictors exhibit strong correlation, conventional regression techniques may encounter difficulties in yielding consistent results. Lasso, however, often selects a single predictor from a set of associated predictors while reducing the rest to zero, therefore streamlining the model with minimal information loss.

Lasso Regression has been effectively utilized across several fields, including as genetics, image processing, and signal processing. In EEG-based mental fatigue detection, it facilitates the identification of critical neural signatures linked to fatigue by extracting pertinent elements from complex brain connection metrics. This results in models that are both more interpretable and have enhanced generalization performance on new data.

2.2.3.10 Recursive Feature Elimination with Correlation Bias Reduction (RFE-CBR-linear)

Recursive Feature Elimination with Correlation Bias Reduction (RFE-CBR-linear) is a supervised feature selection technique aimed at improving machine learning model performance by efficiently identifying informative features and reducing the negative impact of feature collinearity. Traditional Recursive Feature Elimination (RFE) is a widely utilized method that systematically eliminates the least significant features according to the weights determined by a classifier, commonly a Support Vector Machine (SVM) with a linear kernel. However, RFE can be biased when features are correlated, as the weights assigned by the linear SVM may not accurately reflect the true importance of each feature due to redundancy among features [74], [75].

RFE-CBR-linear mitigates this problem by integrating a Correlation Bias Reduction (CBR) approach into the RFE methodology. The procedure starts by training a linear SVM on the complete feature set to derive initial weight coefficients. The weights are subsequently modified to reflect relationships among attributes. This modification entails calculating the correlation matrix of the characteristics and adjusting the weight of each feature inversely relative to its association with other features. This strategy mitigates bias from correlated features, guaranteeing that features are assessed according to their distinct contributions to prediction performance.

The modified weights are employed to prioritize the characteristics, and the feature of least significance is discarded. The procedure is recursive; the SVM is re-trained on the reduced feature set, and the weights are recalibrated and modified for correlation bias in each iteration. This process persists until a certain number of

characteristics is retained or a stopping requirement is fulfilled. The outcome is a sequential enumeration of properties that are relevant and minimally redundant.

In high-dimensional datasets, such as those used in biological signal processing, including electroencephalography (EEG) data, RFE-CBR-linear is very beneficial. EEG data frequently exhibit significant feature correlation due to the intricate relationships among brain impulses. By mitigating the effects of feature correlation, RFE-CBR-linear identifies features that offer distinct and substantial information, hence improving the classifier's capacity to generalize to new data.

A significant advantage of RFE-CBR-linear is the incorporation of feature selection within the model training process, which takes into account the interactions between features and the target variable. Furthermore, by mitigating correlation bias, the technique circumvents the possible drawbacks of conventional RFE, including the removal of relevant characteristics that exhibit connection with others.

2.2.3.11 Recursive Feature Elimination with Correlation Bias Reduction (RFE-CBR-gaussian)

Recursive Feature Elimination with Correlation Bias Reduction using a Gaussian Kernel (RFE-CBR-gaussian) is a sophisticated supervised feature selection technique aimed at improving machine learning efficiency in high-dimensional datasets marked by complex, non-linear relationships and feature correlations. Conventional Recursive Feature Elimination (RFE) is a widely utilized method that progressively eliminates the least significant features according to the weights allocated by a classifier, commonly a Support Vector Machine (SVM). Standard RFE may exhibit bias when correlated features are present, since the weight coefficients might not adequately represent the real significance of each feature due to redundancy[74], [75].

RFE-CBR-gaussian mitigates this disadvantage by including Correlation Bias Reduction (CBR) into the RFE methodology and utilizing a Support Vector Machine (SVM) with a Gaussian (Radial Basis Function) kernel. The Gaussian kernel enables the SVM to identify non-linear relationships between features and the target variable by transforming the input data into a higher-dimensional feature space. This capacity is especially vital in fields such as electroencephalography (EEG) signal processing because the fundamental patterns are intrinsically complex and non-linear.

The CBR component modifies the feature significance ratings to consider correlations between features. During each iteration of the recursive elimination procedure, the algorithm calculates the correlation matrix of the existing feature set. It subsequently adjusts the weights allocated by the SVM by diminishing the impact of associated features, therefore guaranteeing that the selection process prioritizes characteristics that offer distinct and substantial contributions to the model's prediction performance.

The RFE-CBR-gaussian algorithm operates iteratively as follows:

1. **Model Training:** An SVM with a Gaussian kernel is trained on the current feature set.
2. **Weight Adjustment:** The weights obtained from the SVM are adjusted based on the feature correlation matrix to mitigate bias.
3. **Feature Ranking:** Features are ranked according to the adjusted weights.
4. **Feature Elimination:** The least important feature(s) are removed from the feature set.

Steps 1-4 are repeated until a predefined number of features remains or a stopping criterion is met.

By incorporating correlation bias reduction and leveraging the non-linear modelling capabilities of the Gaussian kernel, RFE-CBR-gaussian effectively selects a subset of features that are both highly relevant and minimally redundant. This results in improved model generalization and predictive performance, especially in datasets where feature interactions are complex and non-linear.

2.2.4 Classification Algorithms

The ranked selected features acquired from the feature selection algorithms are utilized to train and evaluate machine learning classifiers aimed at differentiating between resting and fatigue states [76]. Five classifiers are employed in this process: K-Nearest Neighbors (KNN), which assigns class labels based on the majority vote among the nearest neighbors, and Support Vector Machines (SVM) with both linear and Radial Basis Function (RBF) kernels for capturing linear and nonlinear relationships, respectively. Linear Discriminant Analysis (LDA) reduces the data to a lower-dimensional space to maximize class separability, while Random Forest (RF) leverages an ensemble of decision trees to improve classification accuracy and mitigate overfitting. Performance metrics, including accuracy, sensitivity, specificity, and the area under the receiver operating characteristic curve (AUC), are used to assess each classifier. The number of features is systematically adjusted to optimize classification performance, and the results are visualized to illustrate the relationship between feature count and accuracy.

2.2.4.1 k-Nearest Neighbors (kNN):

The k-Nearest Neighbors (kNN) algorithm is a simple yet powerful method used in machine learning, particularly for classification tasks. It is an instance-based or lazy learning algorithm, meaning that instead of explicitly training a model, it uses the data itself to make predictions by referencing specific examples from the training set. Its intuitive nature and effectiveness make it a widely used algorithm, especially as a baseline for more complex models in pattern recognition [76], [77], [78].

The kNN algorithm classifies data points based on their proximity to other labelled points in the feature space. For a given, unlabelled data point, the algorithm identifies the k closest points from the training set known as the “neighbours” based on a defined distance metric. The Euclidean distance is the most commonly used metric, where the straight-line distance between two points is calculated, but other metrics, such as Manhattan distance (sum of absolute differences), Minkowski distance (a generalization that includes both Euclidean and Manhattan), or Cosine similarity, can also be applied depending on the data’s nature and distribution.

The algorithm assigns a label to the new data point by majority voting among the k neighbours. For example, if k is set to 3, the algorithm will classify the point according to the majority label among its three nearest neighbors. The choice of k is crucial: smaller values of k make the model more sensitive to the local structure and can lead to overfitting, as it relies heavily on the closest data points, which might include noise or outliers. Larger values of k reduce this sensitivity, averaging over more points, but this can lead to underfitting, especially if points from different classes are included in the neighborhood.

A distinctive feature of kNN is that it is a non-parametric algorithm, meaning it does not make assumptions about the underlying data distribution. This property makes it flexible and able to capture complex, non-linear decision boundaries, making it suitable for various applications where class boundaries are not easily separable. However, this also makes kNN computationally intensive since, for each prediction, it must calculate distances between the query point and every point in the training dataset. This can make kNN unsuitable for very large datasets, as the complexity for each classification scales with the number of instances in the dataset. To improve efficiency, data structures like KD-trees or Ball trees are often employed, particularly for low-dimensional data, as they allow for faster nearest-neighbor searches by organizing data points in a tree structure.

Preprocessing of the data is often necessary to ensure kNN performs optimally. Since distance calculations are sensitive to feature scales, normalization or standardization of features is important, especially in cases where features differ significantly in magnitude. Additionally, kNN is sensitive to irrelevant or redundant features, as these can distort the distance calculations, reducing the model's accuracy. Dimensionality reduction techniques, such as Principal Component Analysis (PCA), or feature selection methods, can be applied to reduce the impact of irrelevant features and improve model performance.

The kNN algorithm is widely used across fields due to its versatility and simplicity. It is popular in medical diagnostics, where it classifies diseases based on patient characteristics. Despite its limitations, such as its high sensitivity to the choice of k and computational inefficiency with large datasets, kNN remains a foundational algorithm in machine learning and is valuable for both practical applications and as a baseline model for comparison. Its simplicity and interpretability offer an excellent introduction to supervised learning and form the groundwork for understanding more complex algorithms.

2.2.4.2 Support Vector Machine (SVM) with RBF Kernel

The Support Vector Machine (SVM) with a Radial Basis Function (RBF) kernel is an effective and widely used machine learning technique, particularly for classification problems involving non-linearly separable data. Support Vector Machine (SVM) is a supervised learning algorithm designed to identify the ideal hyperplane that effectively distinguishes between classes within the feature space. It operates by converting data into a higher-dimensional space and identifying a decision boundary that maximizes the margin—the distance between the nearest data points (support vectors) from each class and the hyperplane [79], [80].

A linear SVM performs effectively when data is linearly separable. In instances where classes cannot be defined by a linear boundary, the RBF kernel, or Gaussian kernel, proves advantageous. The RBF kernel allows the SVM to identify complex, non-linear correlations in the data by transforming the original feature space into a higher-dimensional space where a linear boundary may be formed.

Mathematically, the RBF kernel function is defined as:

$$K(x_i, x_j) = \exp\left(-\gamma\|x_i - x_j\|^2\right) \quad (5)$$

where x_i and x_j are data points, and γ is a parameter that controls the influence of individual training examples. A larger γ value allows each data point to have a smaller influence radius, resulting in a more complex decision boundary, while a smaller γ value creates a smoother boundary.

The SVM with the RBF kernel incorporates a regularization parameter C , which governs the balance between maximizing the margin and decreasing classification mistakes. A high C value indicates that the model seeks to accurately categorize every point, which may result in overfitting, whereas a lower C value permits certain misclassifications to attain a wider margin, hence enhancing generalization. By adjusting C and γ an RBF SVM can be tailored to perform well across various types of data.

The primary benefits of the RBF kernel SVM are its adaptability to non-linear data and its efficacy in high-dimensional environments. The approach performs effectively even when the feature count exceeds the observation count, rendering it advantageous in domains such as text classification, bioinformatics, and picture recognition. Furthermore, while SVMs concentrate just on support vectors—data points next to the decision boundary—they exhibit greater efficiency compared to some other algorithms, as extraneous data points situated further from the margin do not influence the model's decision boundary.

In conclusion, SVM with an RBF kernel provides an adaptable and resilient classifier for intricate, non-linear data. Its capacity to convert data into a higher-dimensional space and identify appropriate decision boundaries renders it a favored option for classification jobs characterized by difficult class separation. Notwithstanding its computing requirements, the RBF SVM is an essential instrument in machine learning owing to its adaptability and robust efficacy in managing intricate patterns.

2.2.4.3 Linear Discriminant Analysis (LDA)

Linear Discriminant Analysis (LDA) is a widespread classification method in machine learning, especially successful for scenarios where the classes are linearly separable. Linear Discriminant Analysis (LDA) serves as both a classification and dimensionality reduction technique, designed to identify a linear combination of characteristics that optimally distinguishes between two or more classes. The fundamental concept of LDA is to enhance the distinction between several classes while reducing the variation within each class, so establishing a decision boundary that effectively differentiates them.[81], [82].

LDA operates by representing each class in the dataset with a Gaussian distribution, supposing that all classes possess a shared covariance matrix. This assumption streamlines the computations and makes LDA appropriate for linear delineations. The approach identifies a projection that optimizes the inter-class variance (the separation between class means) while minimizes the within-class variance (the dispersion of data points within each class). This projection provides a novel set of characteristics, wherein classes are more distinct and readily separable.

Mathematically, LDA seeks to maximize the ratio of the between-class scatter to the within-class scatter. For two classes, the goal is to project the data onto a line where the means of the two classes are as far apart as possible while maintaining a low variance within each class.

Given a dataset with two classes, LDA aims to find a vector w that maximizes the Fisher criterion, which is defined as the ratio of the between-class scatter to the within-class scatter. This vector w defines the projection direction that best separates the two classes.

Within-Class Scatter Matrix S_w : The within-class scatter matrix measures the variance within each class. For two classes with means μ_1 and μ_2 and covariance matrices Σ_1 and Σ_2 , the within-class scatter matrix is given by:

$$S_w = \Sigma_1 + \Sigma_2 \quad (6)$$

$$\Sigma_1 = \sum_{i \in C_1} (x_i - \mu_1)(x_i - \mu_1)^T \quad (7)$$

$$\Sigma_2 = \sum_{i \in C_2} (x_i - \mu_2)(x_i - \mu_2)^T \quad (8)$$

Between-Class Scatter Matrix S_B : The between-class scatter matrix captures the separation between the class means. For two classes, it is defined as:

$$S_B = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T \quad (9)$$

where μ_1 and μ_2 are the means of classes C_1 and C_2 .

Fisher Criterion: The objective is to find a vector w that maximizes the ratio of between-class variance to within-class variance. The Fisher criterion can be expressed as:

$$J(w) = \frac{w^T S_B w}{w^T S_w w} \quad (10)$$

Maximizing $J(w)$ leads to the direction w that best separates the classes.

Optimal Projection: The optimal w is obtained by solving the generalized eigenvalue problem:

$$S_W^{-1} S_B w = \lambda w \quad (11)$$

where λ represents the eigenvalues. The eigenvector corresponding to the largest eigenvalue provides the optimal direction for maximizing class separation.

A primary advantage of LDA is its computational efficiency, especially with high-dimensional data. LDA excels in datasets when the feature count exceeds the sample size, particularly in text and picture classification. The technique is straightforward to implement and interpretable, as it establishes a distinct decision boundary based on the linear configuration of the data. Moreover, LDA serves as an effective dimensionality reduction technique, particularly advantageous for data visualization in two or three dimensions.

In fact, LDA is frequently employed in applications such as facial recognition, medical diagnostics, and text classification, where it adeptly classifies data points using linear bounds. Its interpretability and little processing expense render it a suitable approach for preliminary dataset exploration and for scenarios where linear correlations among classes are anticipated.

2.2.4.4 Random Forest

Random Forest is an ensemble learning algorithm that enhances the performance of individual decision trees by combining many of them to create a "forest." It builds on the idea that multiple weak models, when combined, can create a strong model. Random Forest is especially effective for classification and regression tasks in complex, high-dimensional datasets and is designed to address the limitations of single decision trees, which tend to overfit and can be sensitive to noisy data. The Random Forest algorithm constructs multiple decision trees during training. Each tree is trained on a random subset of the data generated through bootstrap sampling, where samples are drawn with replacement, meaning some data points may be duplicated in a single tree's subset, while others are excluded. This sampling technique, known as Bagging (Bootstrap Aggregating), introduces diversity among the trees, which reduces correlation and makes the model more resilient to noise. Additionally, Random Forest implements random feature selection during the tree-building process, where each split within a tree is made using a random subset of the available features, further ensuring diversity among trees. This combination of random data samples and random feature subsets allows Random Forest to create a collection of diverse trees that capture different aspects of the data and reduces the chance of any individual tree overfitting [76], [83], [84].

Each decision tree in a Random Forest grows until it meets a stopping criterion, such as reaching a maximum depth or a minimum number of samples per leaf. Unlike traditional decision trees, which are pruned to avoid overfitting, Random Forest trees are typically grown to their full depth. The ensemble nature of the forest mitigates the overfitting tendency of individual trees, as errors in one tree are often compensated by the accuracy of others. When making predictions, Random Forest relies on majority voting for classification tasks or averaging for regression tasks. Each tree in the forest makes an independent prediction, and the final prediction is determined by the class that receives the most votes (in classification) or the average prediction (in regression). This voting mechanism provides a more balanced output, reducing the variance seen in single decision trees and resulting in more stable predictions.

A key advantage of Random Forest is its ability to handle high-dimensional datasets with numerous features. It also provides insights into feature importance, as it records how often each feature is selected for splits across trees. Features that are frequently chosen to make splits are considered more influential, offering a form of feature selection and interpretability. This property is particularly useful in fields like

bioinformatics and medical diagnostics, where understanding feature relevance is important.

Random Forest is widely applied in tasks requiring high accuracy, robustness, and feature analysis. Its ability to handle noisy, high-dimensional data while maintaining accuracy makes it popular in applications like fraud detection, image and text classification, and recommendation systems. The model's resilience to overfitting and capacity to generalize well on new data have cemented its reputation as a reliable and powerful machine learning tool.

2.2.4.5 Support Vector Machine (SVM) with Linear Kernel

Support Vector Machine (SVM) with a Linear Kernel is a powerful and efficient machine learning algorithm primarily used for binary classification tasks. SVMs aim to find the optimal hyperplane that separates data into distinct classes with the widest possible margin. A linear SVM is ideal when classes are linearly separable, meaning that a single straight line (or hyperplane in higher dimensions) can effectively divide the classes. The linear kernel makes this SVM variant particularly suitable for high-dimensional datasets where many features contribute to class separation, such as text classification and gene expression analysis [76], [85], [86].

The objective of a linear SVM is to identify the hyperplane that maximizes the margin, which is the distance between the hyperplane and the closest data points from each class. These closest data points are known as support vectors, and they are critical for defining the hyperplane. By maximizing the margin, the SVM minimizes the risk of misclassifying new examples, enhancing the generalizability of the model to unseen data. Mathematically, given a set of training data points (x_i, y_i) , where x_i represents the features, and y_i is the class label (either +1 or -1 for binary classes), the goal is to find a weight vector w and bias term b that define the hyperplane. The optimal hyperplane maximizes the margin, which is formulated as a convex optimization problem:

$$\min_{w,b} \frac{1}{2} \|w\|^2 \quad (12)$$

subject to the constraint:

$$y_i(w^T x_i + b) \geq 1, \forall i \quad (13)$$

This formulation ensures that each data point is correctly classified with a margin of at least 1. In cases where data points are not perfectly separable, soft margins are introduced by adding a regularization parameter C , which allows some misclassification but penalizes them to balance the trade-off between maximizing the margin and minimizing classification errors. The optimization problem then becomes:

$$\min_{w,b,\xi} \frac{1}{2} \|w\|^2 + C \sum_i \xi_i \quad (14)$$

subject to:

$$y_i(w^T x_i + b) \geq 1 - \xi_i, \quad \xi_i > 0 \quad (15)$$

where ξ_i are slack variables that allow margin violations for points within the margin or on the incorrect side of the hyperplane. The parameter C controls the degree of tolerance for misclassified points. A high C value prioritizes correct classification by allowing fewer margin violations but may lead to overfitting, while a low C allows more margin violations, which can improve generalization.

Linear SVMs are computationally effective, particularly when dealing with high-dimensional datasets. The linear SVM is capable of operating directly in the original feature space, which saves both time and computational resources, in contrast to

SVMs with non-linear kernels, which may necessitate transforming the data into a higher-dimensional space. This effectiveness makes it particularly well-suited for text classification tasks, in which the number of features significantly exceeds the number of samples.

Linear SVMs are restricted to linearly separable or nearly separable problems, despite their advantages. Non-linear kernels (such as polynomial or RBF) are more effective for datasets with complex, non-linear relationships between classes. Nevertheless, linear SVMs provide simplicity, interpretability, and efficiency when classes are approximately linearly separable. The interpretability of the model is a direct result of the direct relationship between the weight vector w and feature prominence. Specifically, features with larger weights have a more significant impact on classification.

In conclusion, the Support Vector Machine with a Linear Kernel is a classification algorithm that is both interpretable and robust, making it an excellent choice for high-dimensional, linearly separable data. It is a superb choice for tasks such as text categorization, image classification, and a variety of biomedical applications that require effective separation between classes, as it strikes a balance between simplicity and performance.

3 EEG Analysis Results

3.1 Classification Results

We present the results of the EEG analysis conducted using eleven advanced feature selection algorithms, each designed to identify the most relevant features from the high-dimensional dataset. These algorithms were evaluated systematically by applying them to five different machine learning classifiers: k-Nearest Neighbors (kNN), Support Vector Machines (SVM) with linear and RBF kernels, Linear Discriminant Analysis (LDA), and Random Forest. Each classifier was configured to process a maximum of 500 features selected by the respective algorithms, ensuring a standardized comparison across methods.

In addition to individual classifier performance, the product of the five classifiers' outcomes was also analyzed to assess the consistency of feature selection results. This approach ensures that the identified features are robust and reliable across different classification paradigms, highlighting their general applicability and relevance to the task of distinguishing between fatigue and rest states. By incorporating multiple classifiers into the evaluation process, we emphasize the importance of feature selection consistency, as variability in results could undermine the interpretability and reliability of the framework. The results provide a comprehensive understanding of the feature selection algorithms' effectiveness and their impact on classification accuracy and robustness.

In figure 4 (page 41) the ILFS feature selection accuracy of the five classifiers is depicted. KNN is the most robust and stable classifier, maintaining high accuracy across all feature ranges. LDA performs well initially with fewer features but struggles as more features are added. RF shows significant fluctuations, indicating high sensitivity to feature selection and potential overfitting. SVM-RBF performs steadily with moderate accuracy, lagging behind KNN. SVM-LIN consistently shows poor performance, suggesting the data may not be linearly separable.

Additionally in figure 5 (page 41) the ILFS feature selection combined accuracy is depicted. The combined Accuracy of the classifiers using the ILFS feature selection algorithm is showing poor performance which degrades more by adding more features.

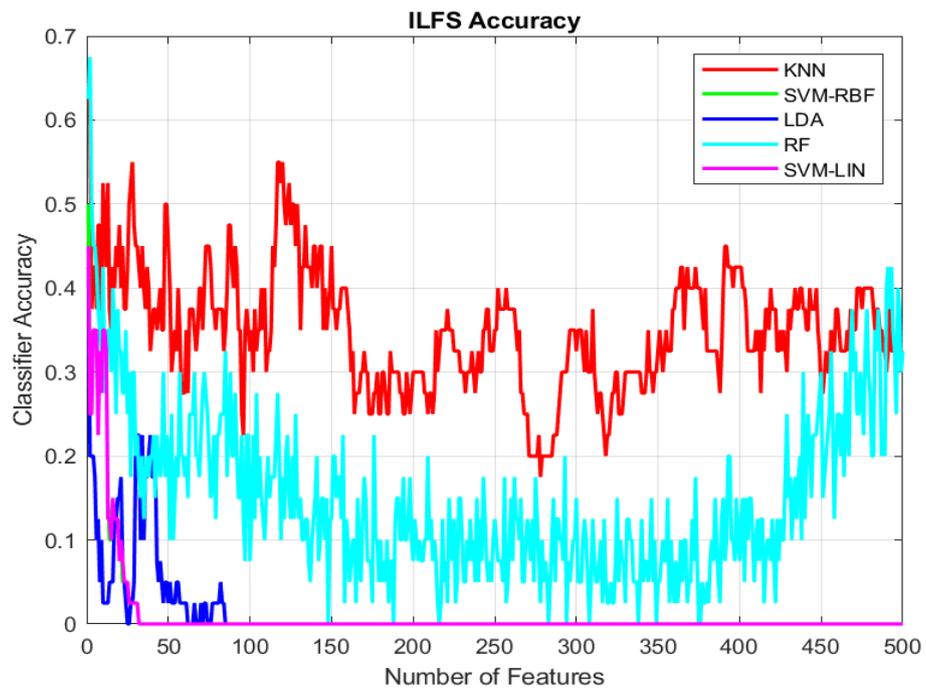


Figure 4: ILFS Accuracy of 5 classifiers

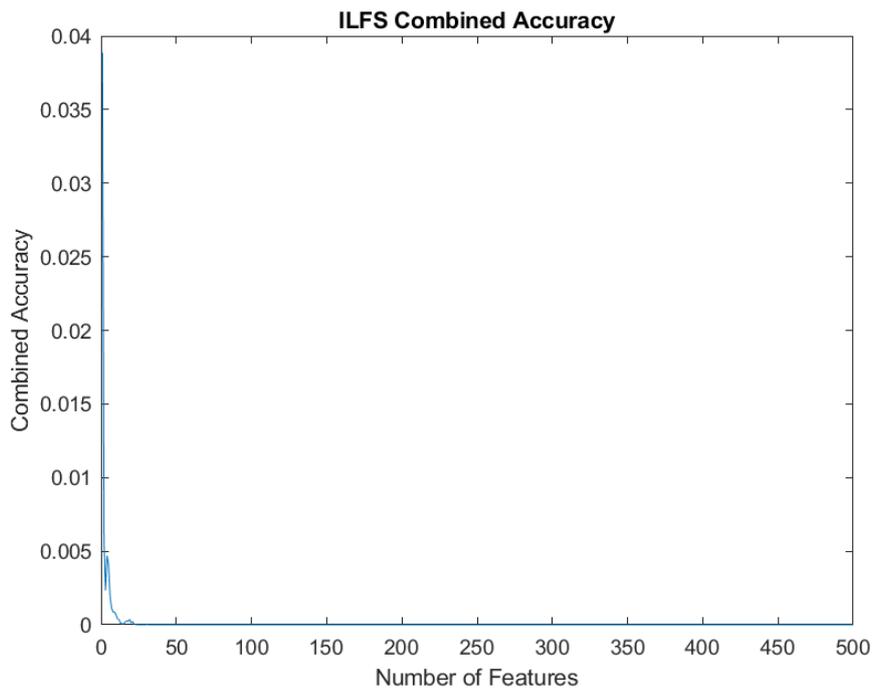


Figure 5: ILFS Product of 5 classifiers accuracy

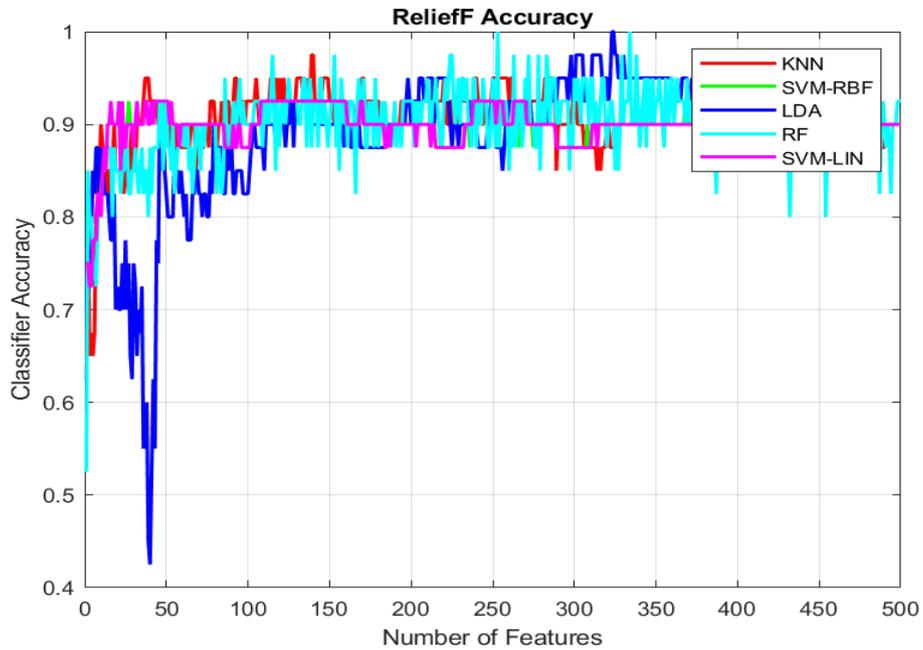


Figure 6: ReliefF Accuracy of 5 classifiers

All classifiers achieve high accuracy (>0.9) as the number of features increases beyond 50, with minimal variations afterward. Initially, LDA and RF show instability but improve significantly with more features. KNN, SVM-RBF, and SVM-LIN maintain consistently high accuracy throughout, demonstrating their robustness. Overall, ReliefF effectively selects features that enhance classifier accuracy, with the differences between classifiers diminishing as more features are added.

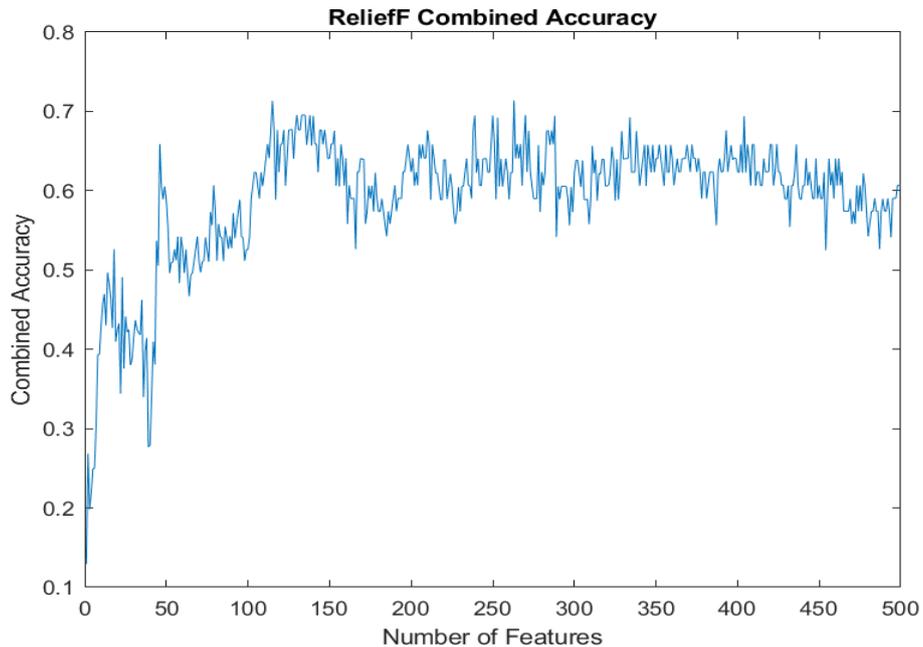


Figure 7: ReliefF Product of 5 classifiers accuracy

This graph (fig 7) shows the combined accuracy of classifiers using ReliefF-selected features. Combined accuracy increases sharply with the first 50 features and stabilizes near 0.6 to 0.7 as more features are added. While there is some fluctuation, the accuracy plateaus after approximately 100 features, indicating that additional features do not significantly improve performance. ReliefF effectively selects a strong subset of features early, with diminishing returns as more features are included.

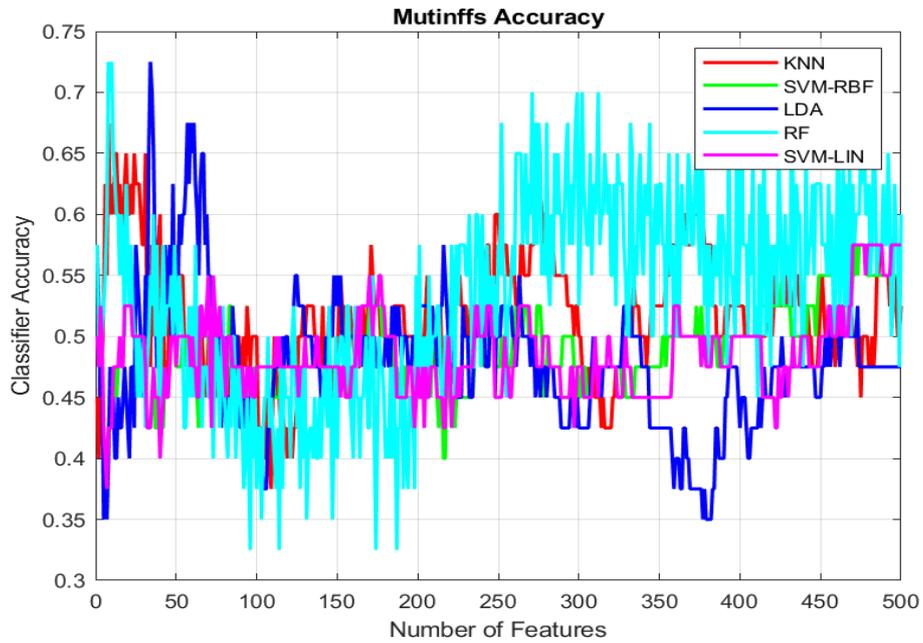


Figure 8: Mutinffs Accuracy of 5 classifiers

Overall, performance is inconsistent across all classifiers, with significant fluctuations in accuracy as the number of features increases. KNN and SVM-RBF show relatively stable but moderate performance. LDA and RF demonstrate high variability, with RF having occasional peaks but generally unstable behavior. SVM-LIN remains consistently low, suggesting poor adaptation to the selected features. The results indicate that MutInfFS struggles to produce informative features, leading to varying classifier performance across the feature range.

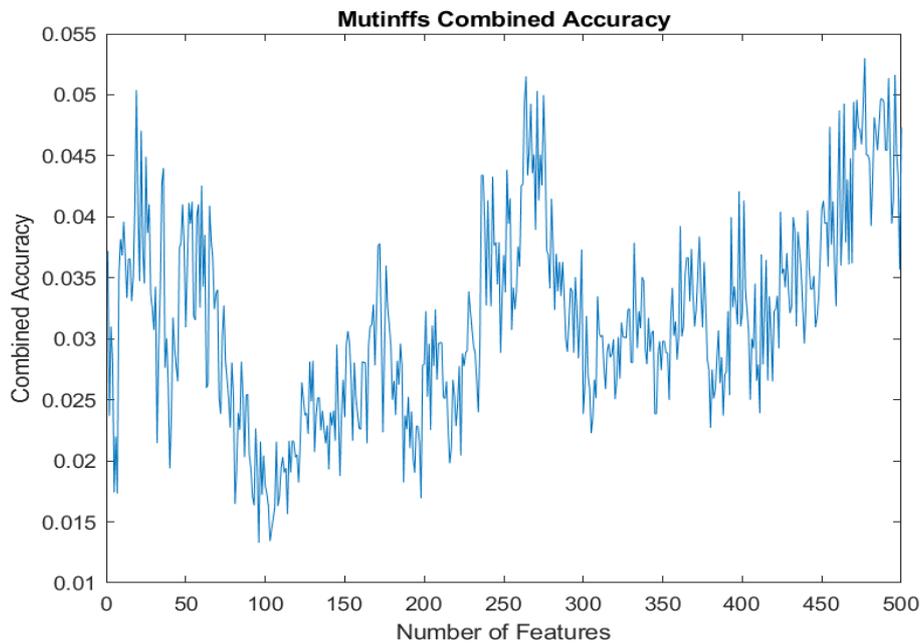


Figure 9: Mutinffs Product of 5 classifiers accuracy

The combined accuracy exhibits substantial fluctuations, with no clear stabilization trend. Combined accuracy starts relatively high, drops around 100–250 features, and gradually increases again toward the end, peaking slightly near 500 features. These variations suggest that the feature selection process introduces instability, with limited subsets contributing meaningfully to classifier performance. The results indicate that MutInfFS does not consistently select the most effective features across the range.

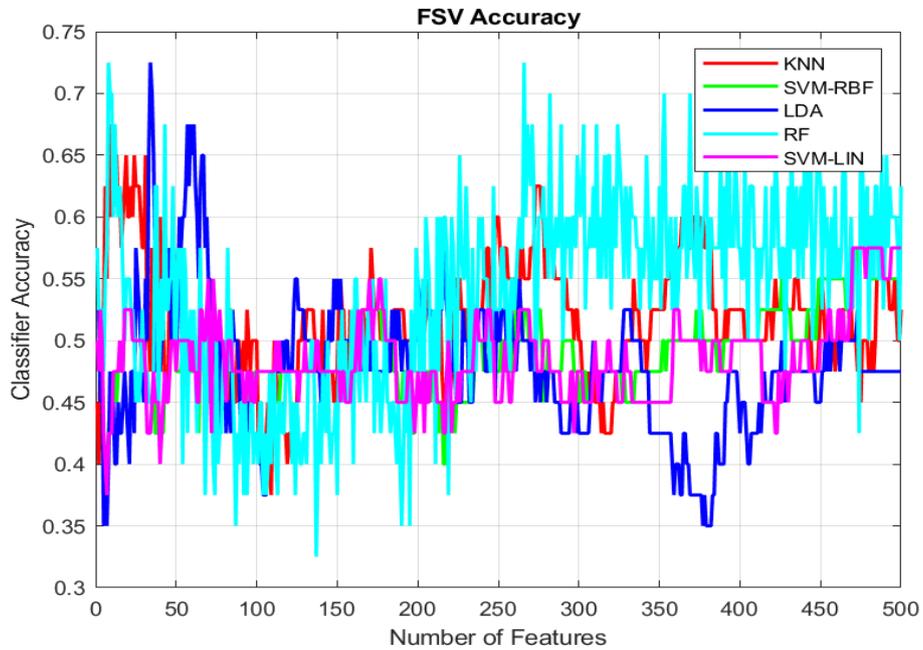


Figure 10: FSV Accuracy of 5 classifiers

The figure 10 displays the accuracy of various classifiers using FSV-selected features. The results are highly variable, with RF showing the most significant fluctuations and occasional peaks in accuracy, while KNN and SVM-RBF maintain moderate and more stable performance. LDA demonstrates instability, particularly as the number of features increases, and SVM-LIN exhibits consistently low accuracy throughout the range. Overall, FSV struggles to produce a consistently effective feature set, leading to fluctuating performance across all classifiers and feature ranges.

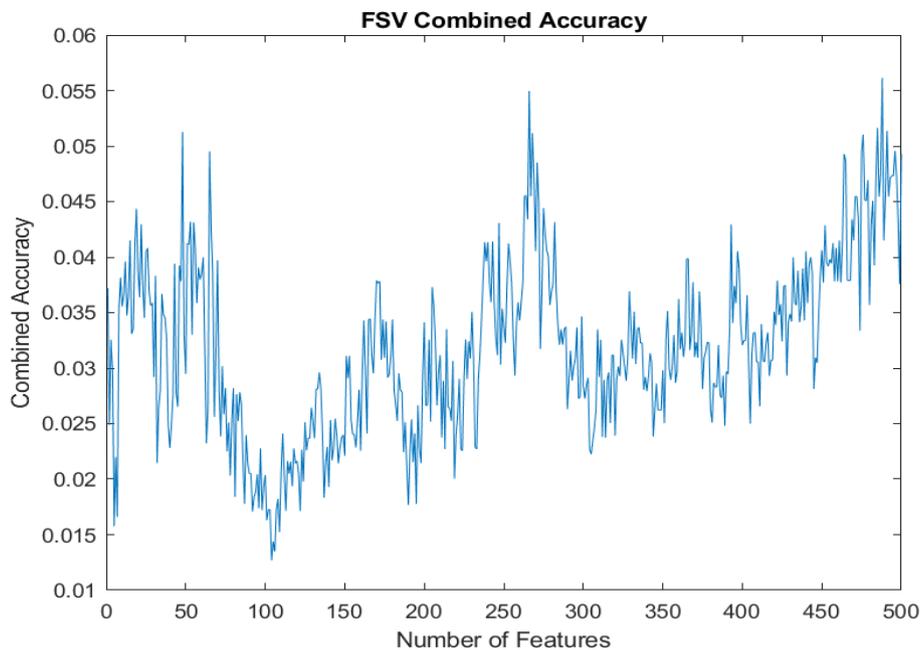


Figure 11: FSV Product of 5 classifiers accuracy

The combined accuracy fluctuates significantly throughout, with noticeable peaks and troughs, particularly around 50 to 150 features. There is a general upward trend in combined accuracy beyond 300 features, indicating gradual improvement. However, the overall performance remains inconsistent, suggesting that FSV-selected features might not consistently capture the most relevant information across the entire feature range.

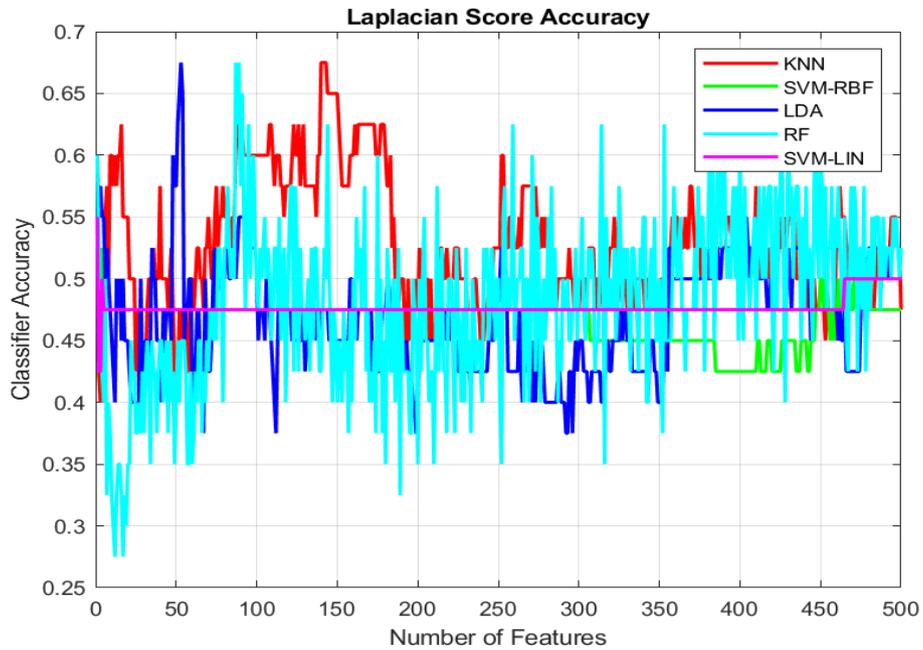


Figure 12: Laplacian Score Accuracy of 5 classifiers

This graph (fig 12) shows the accuracy of classifiers using Laplacian Score-selected features. KNN achieves relatively higher and more stable accuracy compared to other classifiers, with moderate fluctuations. RF demonstrates significant variability, occasionally peaking but frequently dropping. SVM-RBF and LDA show consistently moderate performance with less fluctuation. SVM-LIN remains nearly constant at a lower accuracy level throughout. Overall, the Laplacian Score method yields moderately effective features for KNN, but the performance of other classifiers suggests the selected features may not generalize well across different models.

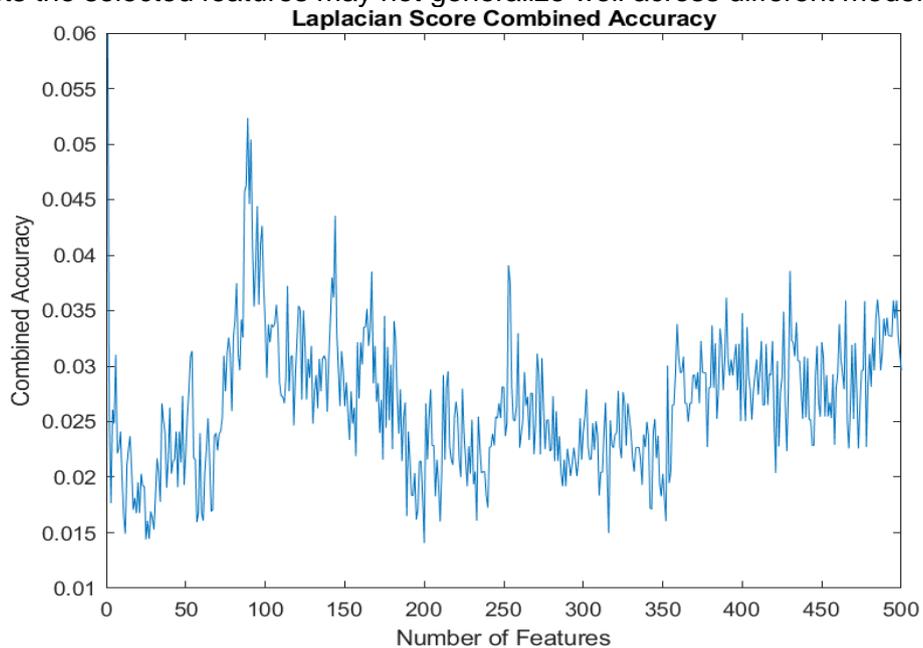


Figure 13: Laplacian Score Product of 5 classifiers accuracy

This graph (fig 13) illustrates the combined accuracy of classifiers using Laplacian Score-selected features. The accuracy peaks early, around 50–100 features, and then fluctuates with a slight downward trend before stabilizing at a modest level beyond 300 features. These fluctuations suggest that the Laplacian Score method captures relevant features in the early stages but struggles to maintain consistency as more features are added. This indicates that the most informative features are likely concentrated in the initial subset of features.

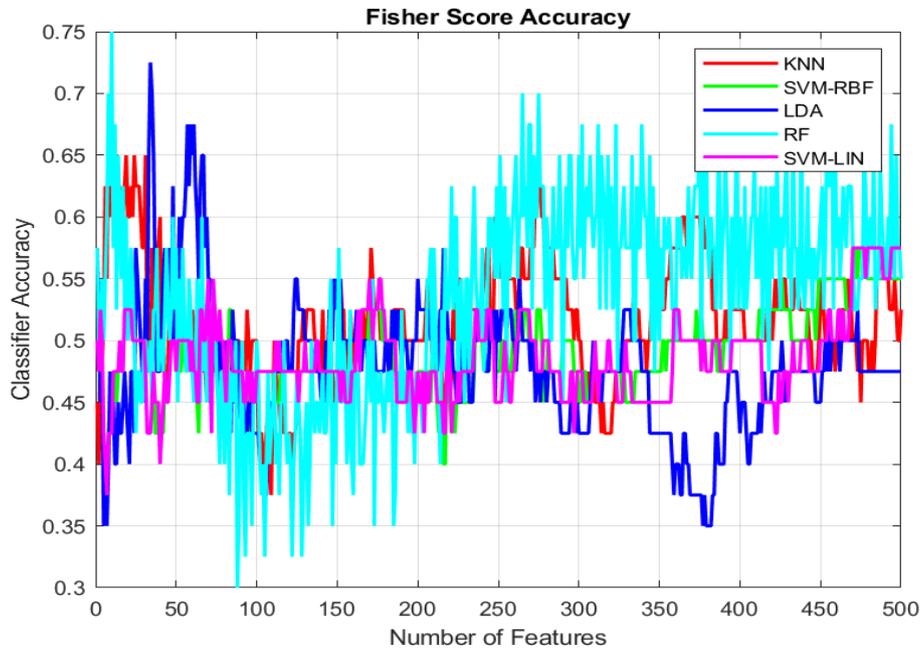


Figure 14: Fisher Score Accuracy of 5 classifiers

This graph (fig 14) represents classifier accuracy using Fisher Score-selected features. KNN and RF show moderate variability, with KNN maintaining relatively stable and higher performance. RF fluctuates significantly, peaking sporadically but often dropping. SVM-RBF and LDA display consistent but average performance, with little improvement as features increase. SVM-LIN remains steady at a low accuracy throughout. The Fisher Score method produces effective features for KNN but shows limited benefit for other classifiers.

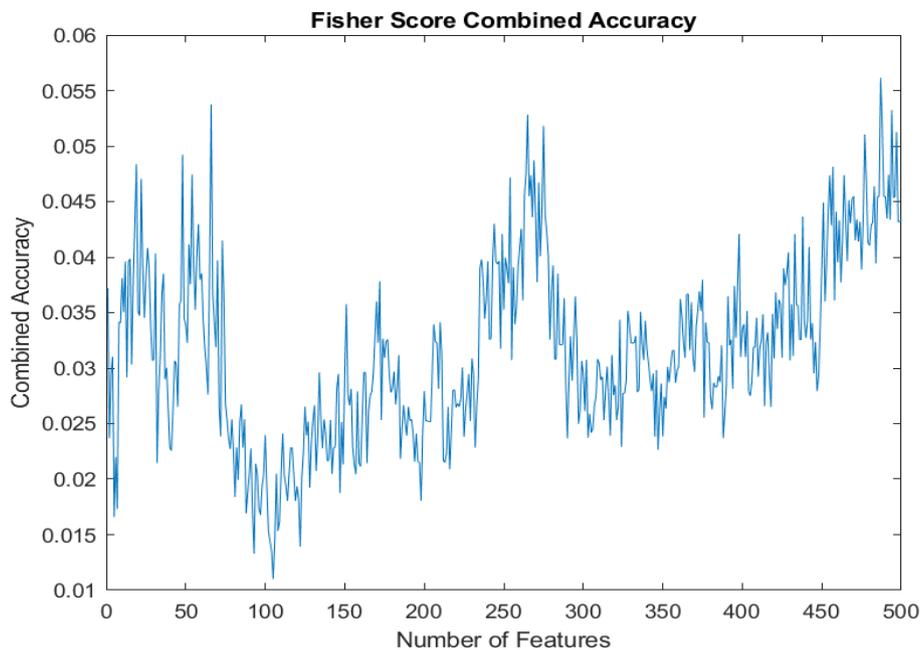


Figure 15: Fisher Score Product of 5 classifiers accuracy

This graph (fig 15) shows the combined accuracy of classifiers using Fisher Score-selected features. The combined accuracy fluctuates significantly in the early feature range (0–100), decreases around 150–250 features, and gradually improves as more features are added, peaking near the end. While the upward trend after 300 features indicates some improvement, the variability suggests that Fisher Score-selected features do not consistently enhance classifier performance across the range.

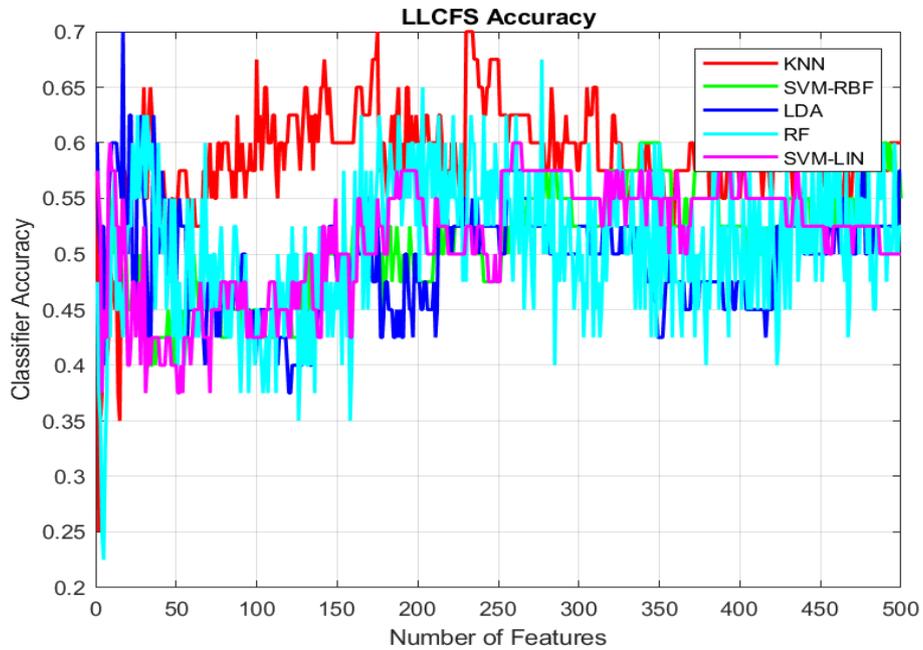


Figure 16: LLCFS Accuracy of 5 classifiers

This graph (fig 16) shows classifier accuracy using LLCFS-selected features. KNN achieves the highest and most consistent accuracy, stabilizing above 0.6 with minimal fluctuations as the number of features increases. RF exhibits significant variability but stabilizes moderately over time. SVM-RBF and LDA maintain relatively steady performance at lower accuracy levels, while SVM-LIN consistently performs poorly. LLCFS appears effective at selecting features that benefit KNN, but its generalizability to other classifiers is limited, as reflected in the modest and variable accuracy of the other models.

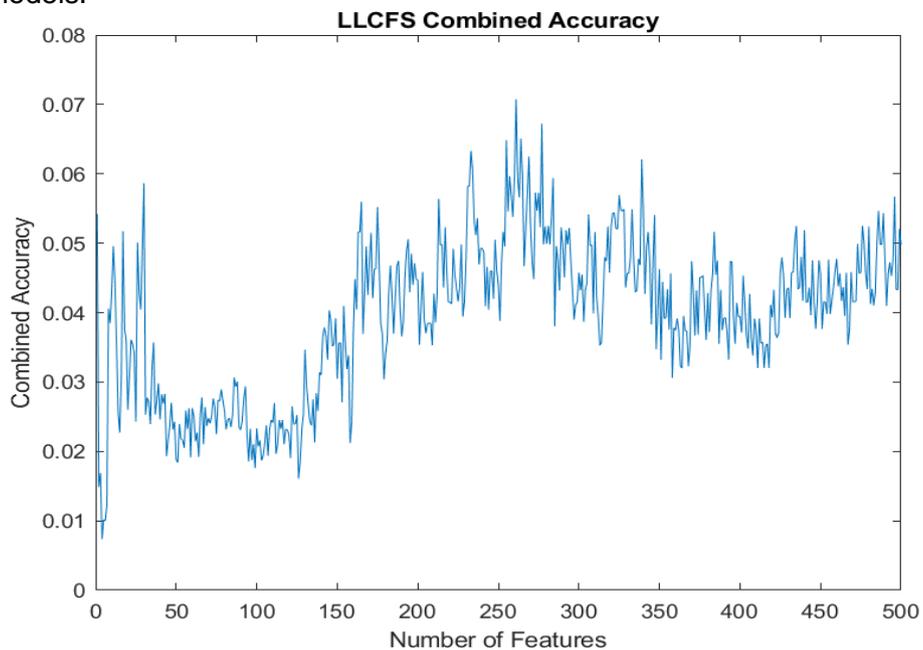


Figure 17: LLCFS Product of 5 classifiers accuracy

This graph (fig17) illustrates the combined accuracy of classifiers using LLCFS-selected features. The combined accuracy starts with significant fluctuations in the early feature range (0–100), gradually improves, and peaks around 200–300 features. Beyond this point, the accuracy stabilizes but shows occasional variability, maintaining a moderate level toward the end. These trends suggest that LLCFS effectively identifies relevant features in the middle range, but the improvements diminish with the inclusion of additional features.

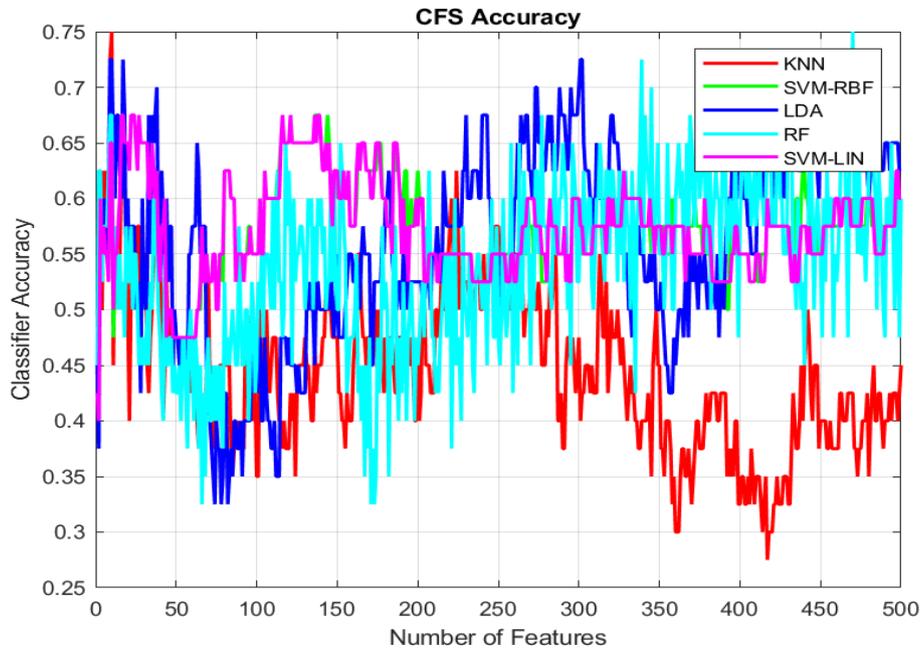


Figure 18: CFS Accuracy of 5 classifiers

This graph (fig18) represents classifier accuracy using CFS-selected features. SVM-LIN and SVM-RBF exhibit relatively stable accuracy, with SVM-LIN achieving moderate performance throughout. RF shows significant fluctuations, with occasional peaks but inconsistent behavior overall. KNN experiences a decline in accuracy as the number of features increases, particularly after 300 features, while LDA displays instability and moderate accuracy. The results suggest that CFS-selected features provide limited and inconsistent benefits across classifiers.

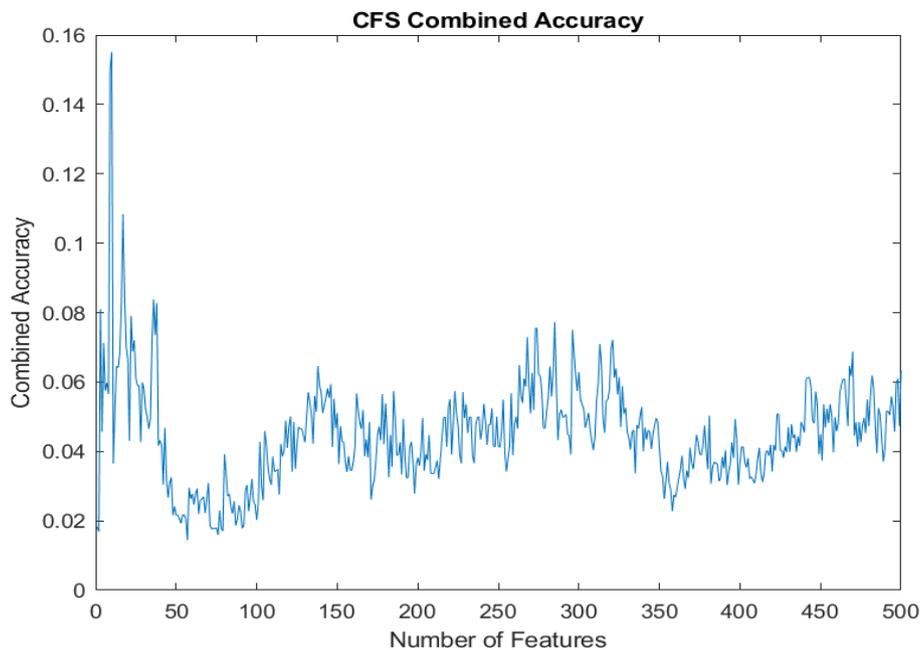


Figure 19: CFS Product of 5 classifiers accuracy

This graph (fig19) illustrates the combined accuracy of classifiers using CFS-selected features. The combined accuracy peaks early, around 0–50 features, and then declines significantly, stabilizing at a much lower level beyond 100 features with minor fluctuations. The early peak suggests that CFS is effective at selecting a small subset of highly relevant features, but its ability to maintain performance diminishes as more features are added, likely due to the inclusion of less informative or redundant features.

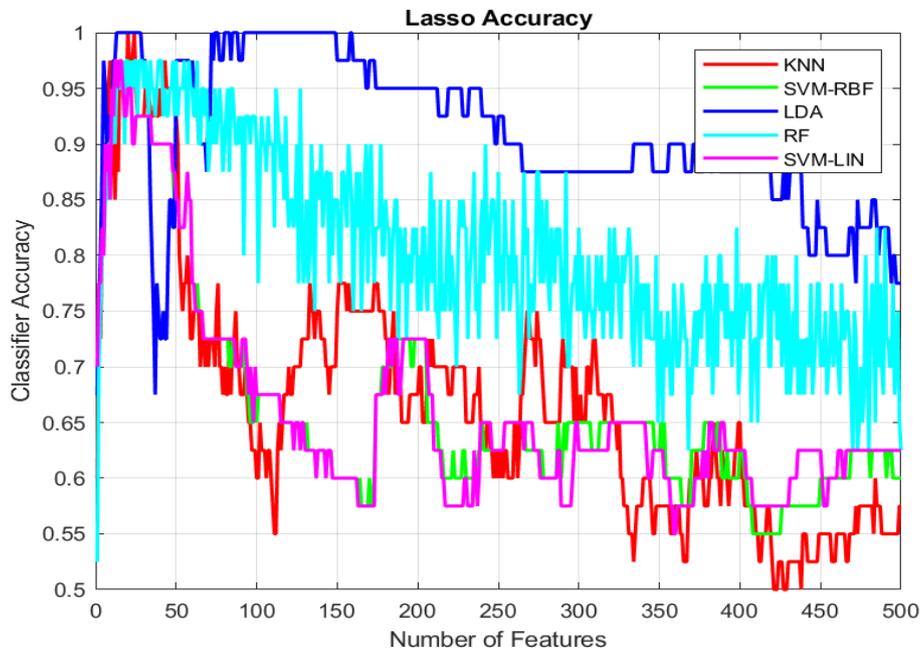


Figure 20: Lasso Accuracy of 5 classifiers

This graph (fig 20) presents classifier accuracy using Lasso-selected features. LDA achieves the highest and most stable accuracy, maintaining near-perfect performance (>0.9) across the range of features. RF exhibits significant variability but generally achieves high accuracy with occasional drops. KNN, SVM-RBF and SVM-LIN show moderate accuracy, with a noticeable decline as the number of features increases. Overall, Lasso demonstrates strong effectiveness in all classifiers, with performance varying as additional features are added.

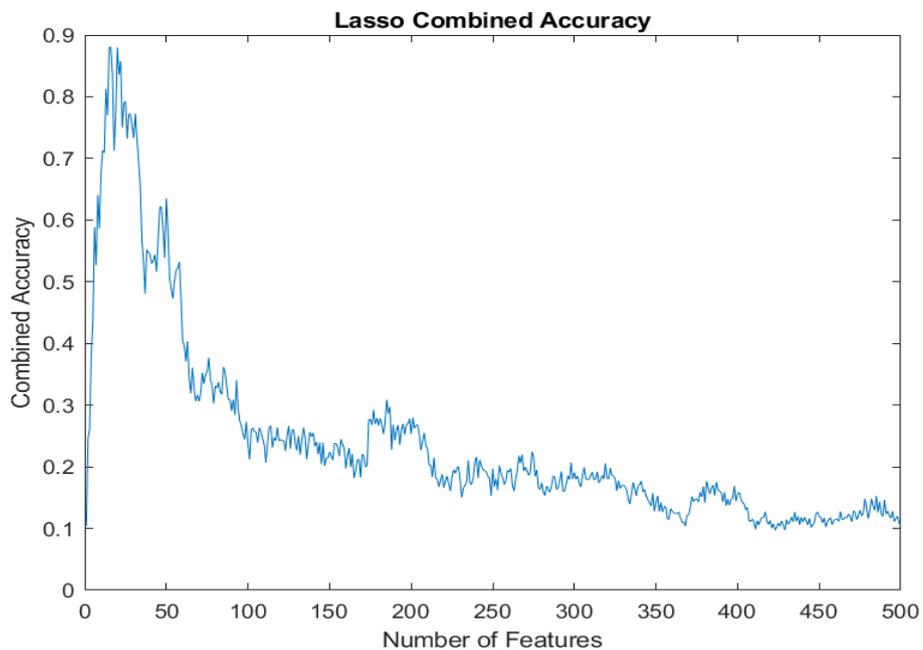


Figure 21: Lasso Product of 5 classifiers accuracy

Lasso achieves the best combined accuracy among all methods, with a peak near 0.9 in the early feature range (up to ~50 features). The performance gradually declines as more features are added, stabilizing at a lower level after 200 features. This highlights Lasso's strength in selecting a small, highly relevant subset of features that significantly enhances classifier performance in the initial stages.

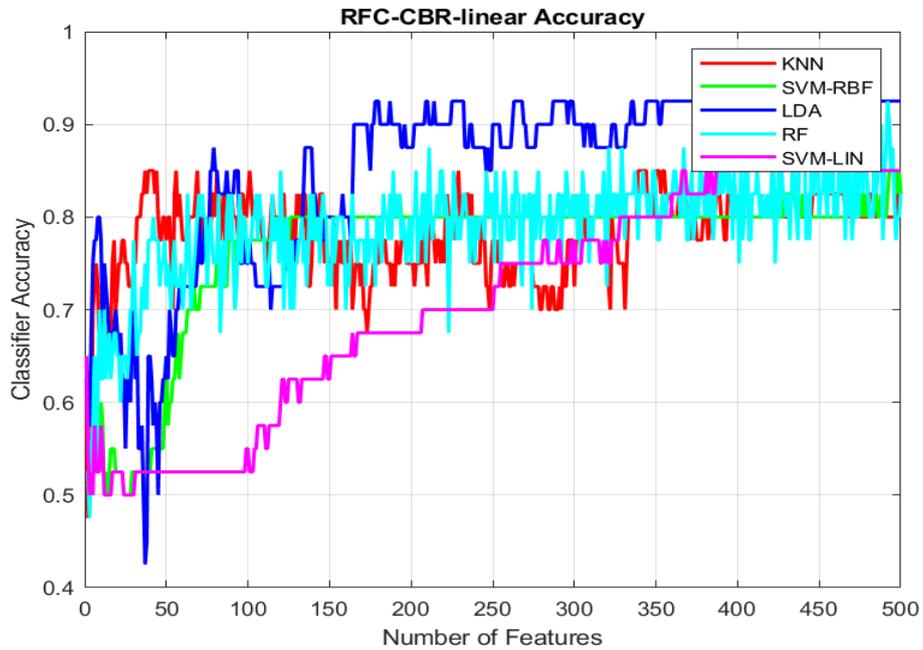


Figure 22: RFC-CBR-linear Accuracy of 5 classifiers

This graph (fig 22) shows classifier accuracy using RFC-CBR-linear-selected features. LDA achieves the highest accuracy, stabilizing near 0.9 as the number of features increases. KNN and RF follow closely with consistent performance around 0.8 after an initial period of variability. SVM-RBF maintains moderate accuracy with slight fluctuations, while SVM-LIN starts lower but steadily improves. Overall, RFC-CBR-linear demonstrates effective feature selection, leading to strong and stable performance for most classifiers.

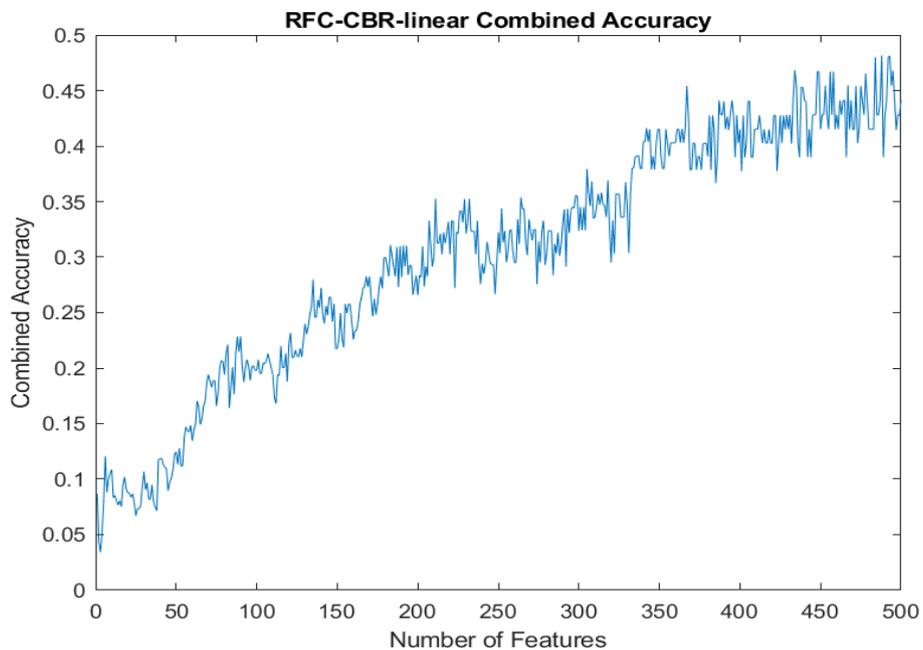


Figure 23: RFC-CBR-linear Product of 5 classifiers accuracy

This graph (fig 23) illustrates the combined accuracy of classifiers using RFC-CBR-linear-selected features. Combined accuracy shows a steady and consistent increase as the number of features grows. The trend indicates that this method effectively selects progressively useful features, contributing to improved performance across classifiers. RFC-CBR-linear demonstrates strong scalability, with no significant decline in accuracy as more features are added.

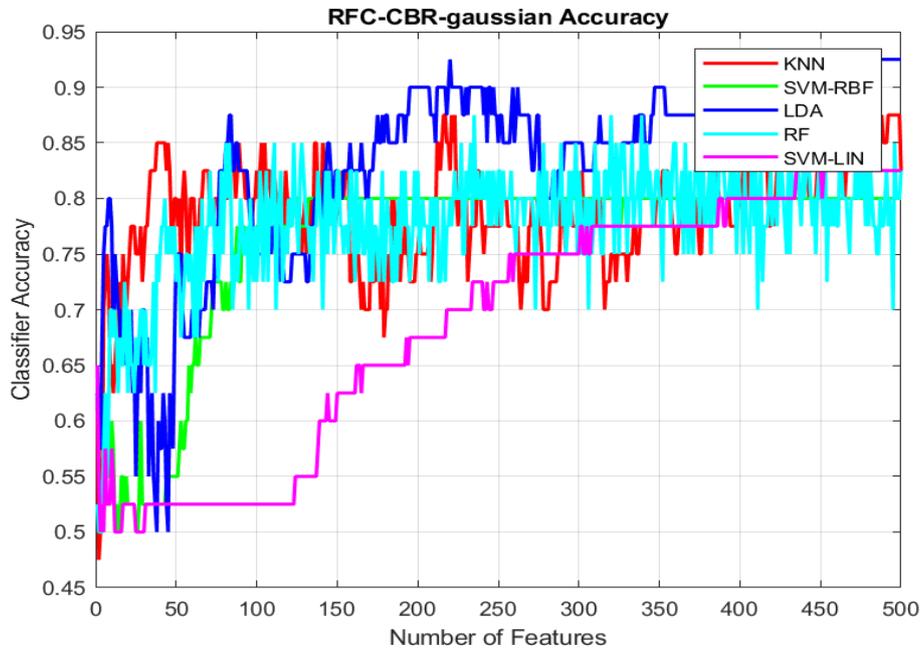


Figure 24: RFC-CBR-gaussian Accuracy of 5 classifiers

This graph (fig 24) shows classifier accuracy using RFC-CBR-gaussian-selected features. LDA achieves the highest and most stable performance, consistently maintaining accuracy near 0.9. KNN and RF also perform well, stabilizing at high accuracy levels (around 0.8–0.85) after initial fluctuations. SVM-RBF shows moderate performance with slight variability, while SVM-LIN starts low but steadily improves. Overall, RFC-CBR-gaussian effectively selects features that enhance classifier performance.

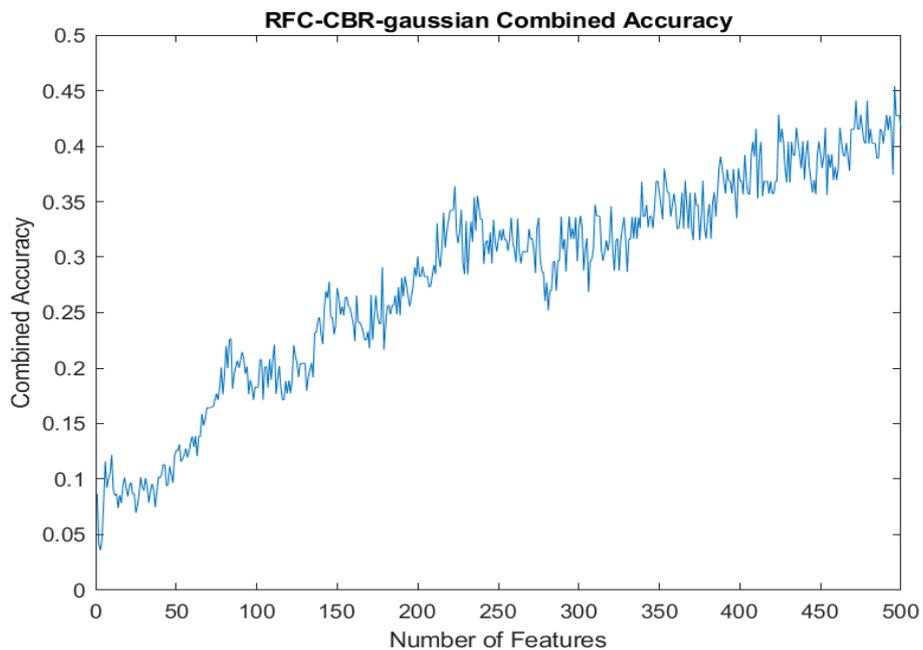


Figure 25: RFC-CBR-gaussian Product of 5 classifiers accuracy

This graph (fig 25) illustrates the combined accuracy of classifiers using RFC-CBR-gaussian -selected features. Combined accuracy shows a steady and consistent increase as the number of features grows. The trend indicates that this method effectively selects progressively useful features, contributing to improved performance across classifiers.

The analysis of 11 feature selection methods across 5 classifiers (KNN, SVM-RBF, SVM-LIN, LDA, and RF) reveals varying degrees of effectiveness in selecting informative features. **Lasso** consistently demonstrated exceptional performance, achieving the highest combined accuracy early (~0.9) with small feature subsets, making it ideal for scenarios requiring a concise selection of relevant features. **ReliefF** also provided robust results, enabling high classifier accuracy (>0.9) for most models after selecting 50 features and achieving combined accuracy around 0.6. **RFC-CBR-linear** and **RFC-CBR-gaussian** methods showed strong scalability, with steadily improving combined accuracy, reaching approximately 0.45 at 500 features, and excellent performance for LDA and RF classifiers. In contrast, methods like ILFS, MutInfFS, and FSV struggled to maintain consistent performance, reflected in low combined accuracy and fluctuations in classifier effectiveness.

3.2 Hyperparameters Optimization Results

In order to optimize the classifiers hyperparameters we used the Bayesian optimization technique. **Bayesian Optimization** is an efficient technique for optimizing black-box functions that are expensive to evaluate and have no known analytical form. It uses a surrogate model, often a Gaussian Process, to approximate the objective function and an acquisition function to decide the next set of parameters to evaluate. This approach balances exploration of new parameter spaces and exploitation of promising areas, enabling it to find optimal solutions with fewer evaluations compared to methods like grid or random search. It is particularly effective for hyperparameter tuning in machine learning, offering a systematic and resource-efficient way to achieve better model performance.

In this study, we used five classifiers and optimized their key hyperparameters to improve performance. For **k-Nearest Neighbors (kNN)**, we optimized the number of neighbors (NumNeighbors) and the distance metric (Distance) to find the best balance between local and global decision boundaries. For **Support Vector Machines (SVM)** with an RBF kernel, we tuned the regularization parameter (BoxConstraint) and kernel scale (KernelScale) to enhance the model's ability to handle non-linear separable data. The **Linear Discriminant Analysis (LDA)** classifier involved optimizing regularization parameters such as Delta and Gamma to manage overfitting and ensure robust covariance estimation.

In the case of **Random Forest (RF)**, we optimized the number of learning cycles (NumLearningCycles) and the minimum leaf size (MinLeafSize) to balance model complexity and generalization. Finally, for **SVM with a linear kernel**, we focused on optimizing the regularization parameter (BoxConstraint) to control the trade-off between margin maximization and classification error. These hyperparameter optimization steps ensured that each classifier operated at its best configuration for the given dataset.

Additionally, we used the parallel computing toolbox of mathworks matlab® which enables parallel evaluations, leveraging multicore processors or clusters to accelerate optimization.

Table 1: Classifiers Optimization

Classifier	Optimization Hyperparameters	Optimization Technique
KNN	NumNeighbors, Distance	Bayesian
SVM RBF	BoxConstraint, KernelScale	Bayesian
LDA	Delta, Gamma	Bayesian
RF	NumLearningCycles, MinLeafSize	Bayesian
SVM Linear	BoxConstraint	Bayesian

In Figure 26 (page 54) The ILFS Accuracy of 5 Optimized classifiers is depicted. The ILFS Optimized Accuracy graph shows that classifier performance remains relatively stable across the range of features, with KNN, RF, and LDA maintaining moderate accuracy around 0.5. SVM-RBF lags slightly behind with lower consistency, while SVM-LIN performs poorly. The optimized ILFS method appears to lack strong discriminatory power for selecting highly effective features, as indicated by the lack of significant improvement in accuracy even with an increasing number of features. This suggests limited effectiveness in optimizing performance across classifiers.

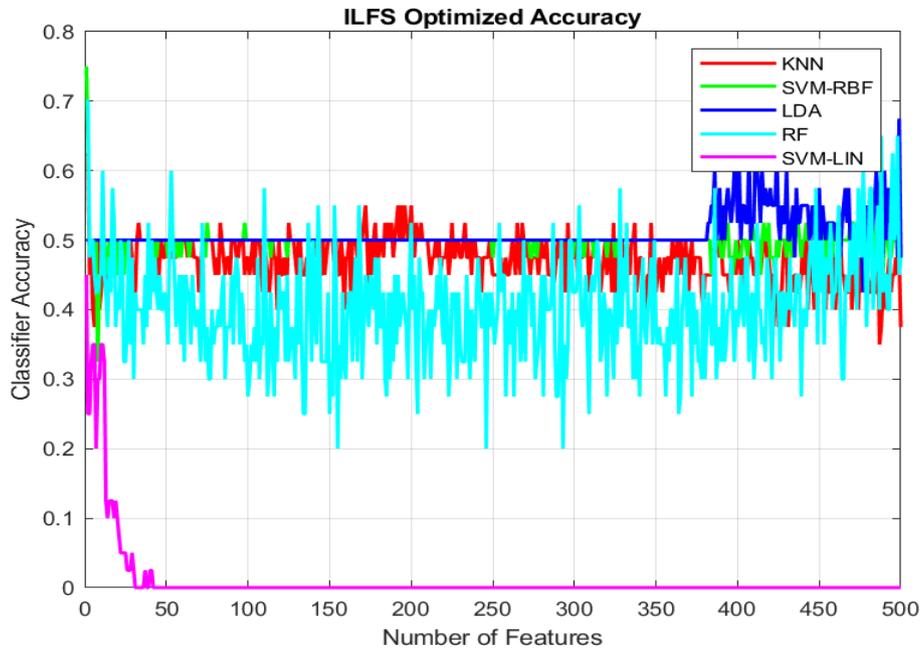


Figure 26: ILFS Accuracy of 5 Optimized classifiers

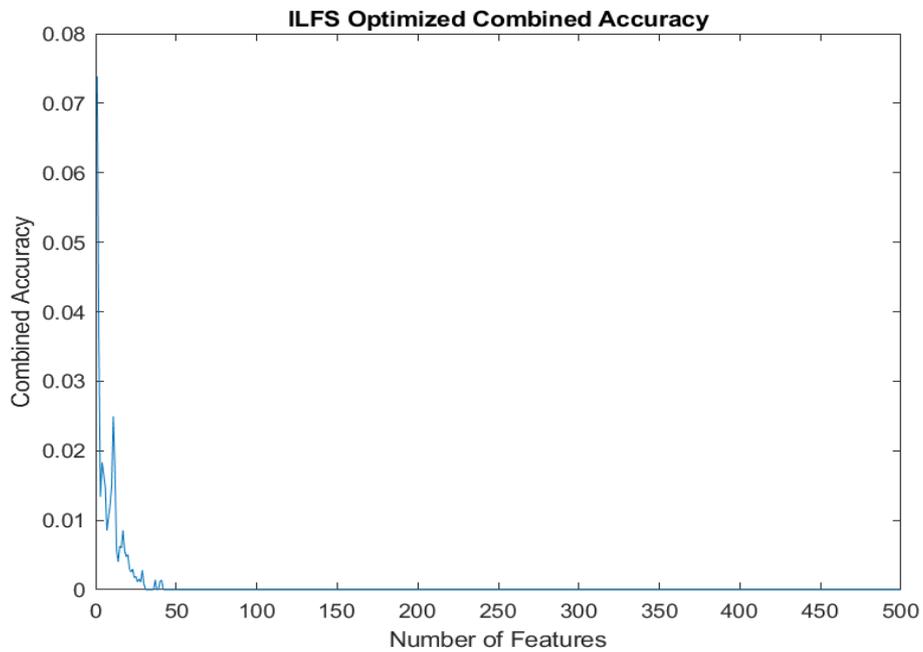


Figure 27: ILFS Product of 5 Optimized classifiers accuracy

The ILFS Optimized Combined Accuracy graph shows that combined accuracy is extremely low, peaking briefly around the first few features (~50 features) and then quickly dropping to near zero. This indicates that the optimized ILFS method fails to effectively enhance the overall performance of the classifiers as the number of features increases. The inability to sustain or improve combined accuracy suggests that the selected features contribute minimally to the overall classification quality, highlighting the limitations of this feature selection approach. The optimization process for ILFS slightly stabilized individual classifier accuracy but did not lead to meaningful improvements in overall performance. The non-optimized ILFS achieved marginally better combined accuracy in the initial feature range, making it the slightly better choice between the two.

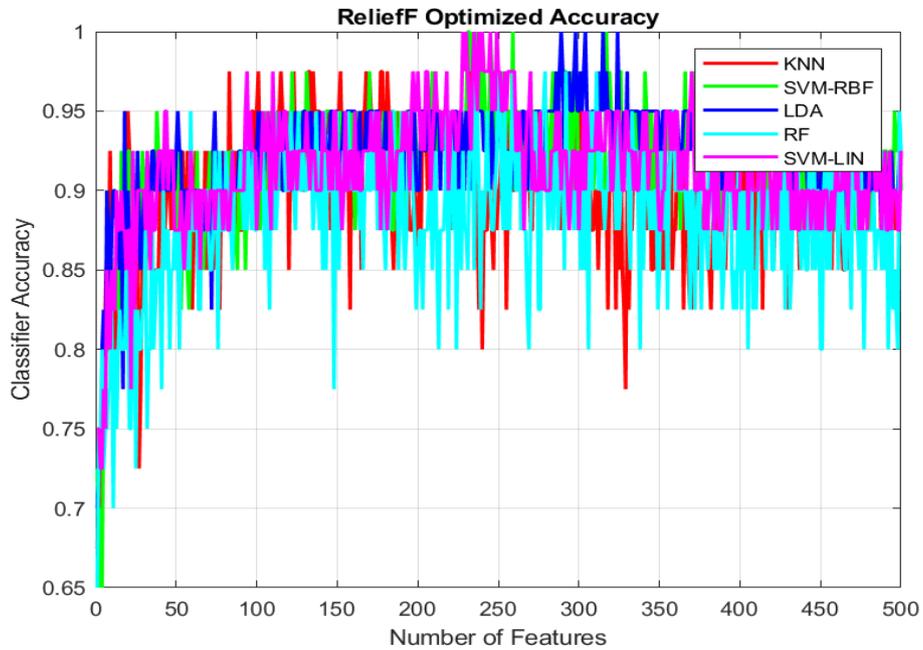


Figure 28: ReliefF Accuracy of 5 Optimized classifiers

The optimized ReliefF method demonstrates significant improvements over the non-optimized version in terms of classifier stability and overall performance. In the non-optimized version, classifiers like KNN, RF, and LDA steadily improved after selecting around 50 features, achieving accuracies above 0.9, while RF exhibited some fluctuations and SVM-LIN underperformed. The optimized ReliefF method not only reduced variability across classifiers but also ensured consistently high accuracy (near or above 0.9) for all models, including significant improvements for SVM-LIN. The stability of RF's performance also increased in the optimized version, making it more robust across the feature range.

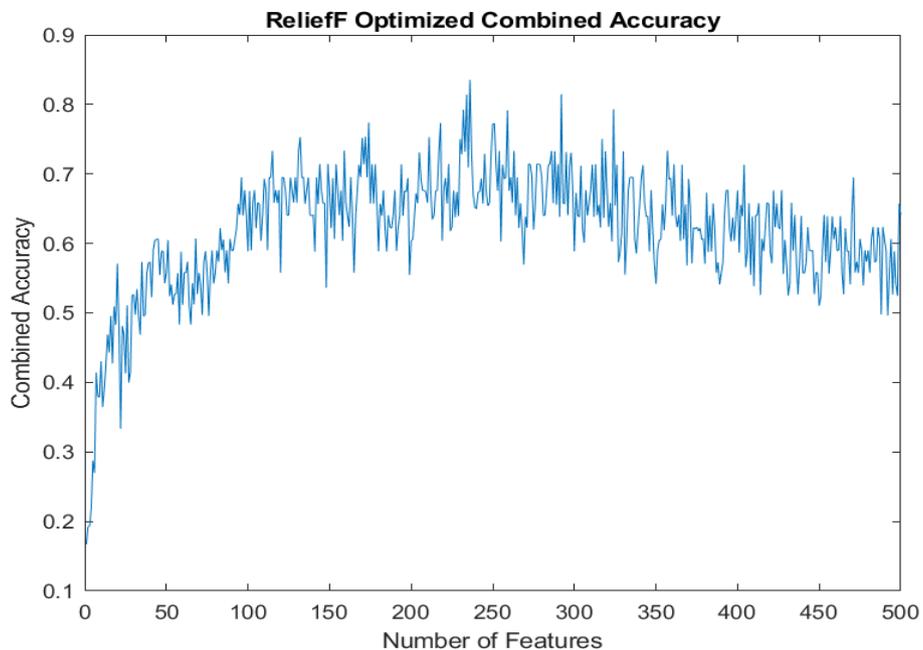


Figure 29: ReliefF Product of 5 Optimized classifiers accuracy

The optimized version of the combined accuracy exhibits a more gradual and consistent increase in accuracy as features are added, with reduced fluctuations and a more stable performance across the feature range. This indicates that the optimization process effectively refines the feature selection, ensuring the inclusion of more relevant and discriminative features while minimizing noise.

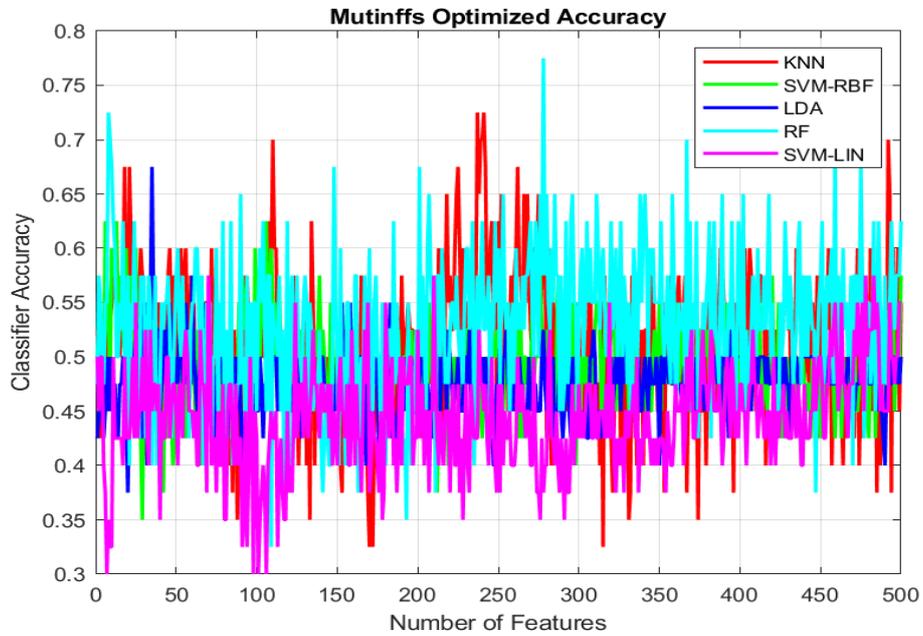


Figure 30: Mutinffs Accuracy of 5 Optimized classifiers

The MutInffS Optimized Accuracy graph (fig 30) shows modest performance across all classifiers, with no significant improvement compared to the non-optimized version. Classifier accuracy remains clustered around 0.4 to 0.6 for most models, with RF and KNN exhibiting occasional spikes but no sustained improvements. SVM-LIN and SVM-RBF maintain stable yet lower accuracies, while LDA shows moderate fluctuations. The optimization process appears to have slightly stabilized performance across features, but it fails to achieve significant gains in accuracy or to enhance overall performance for the classifiers. This suggests that the optimized MutInffS method struggles to identify highly relevant features that could lead to meaningful improvements in classifier accuracy.

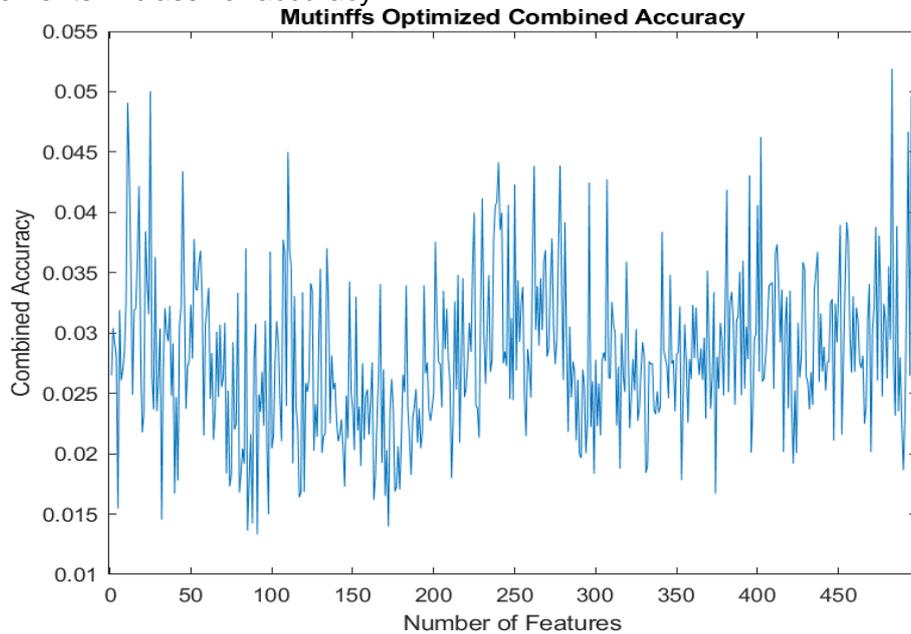


Figure 31: MutInffS Product of 5 Optimized classifiers accuracy

The MutInffS Optimized Combined Accuracy graph (fig 31) reveals that the optimization process has not significantly enhanced the combined accuracy compared to its non-optimized counterpart. Combined accuracy fluctuates heavily across the feature range, remaining relatively low, with values clustered around 0.02 to 0.04. There is no clear trend of sustained improvement or stability. The overall impact of optimization is minimal, reflecting the method's limited ability to consistently identify and prioritize highly relevant features.

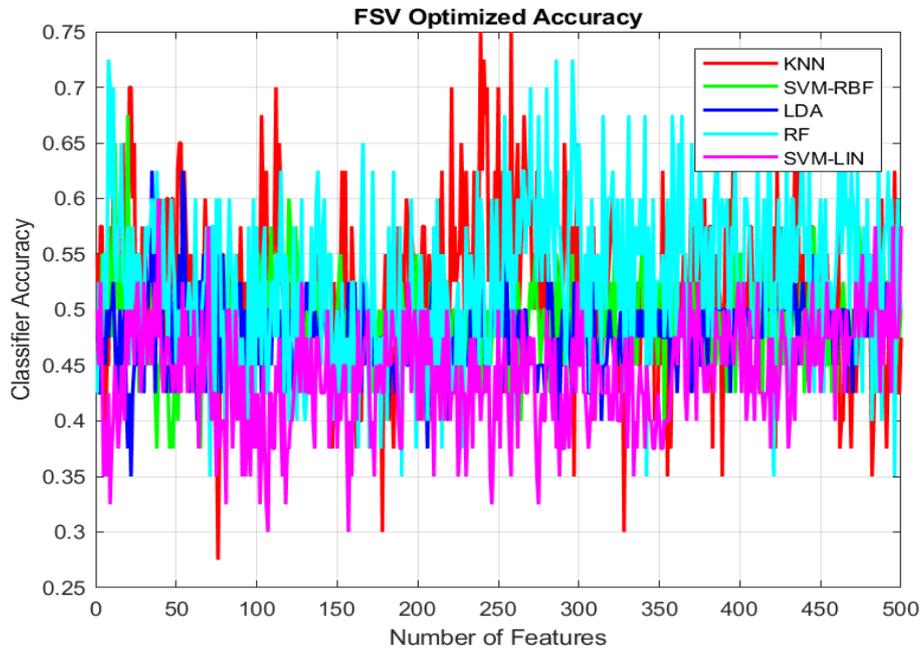


Figure 32: FSV Accuracy of 5 Optimized classifiers

The FSV Optimized Accuracy graph (fig 32) shows modest improvements in classifier stability compared to the non-optimized version, though overall accuracy remains moderate. Classifier performance generally hovers between 0.4 and 0.6, with occasional spikes. Although optimization reduces some fluctuations in accuracy, it does not significantly enhance overall performance across classifiers. This indicates that the optimization process has limited impact on making FSV a more effective feature selection method, as variability persists and overall accuracy remains constrained.

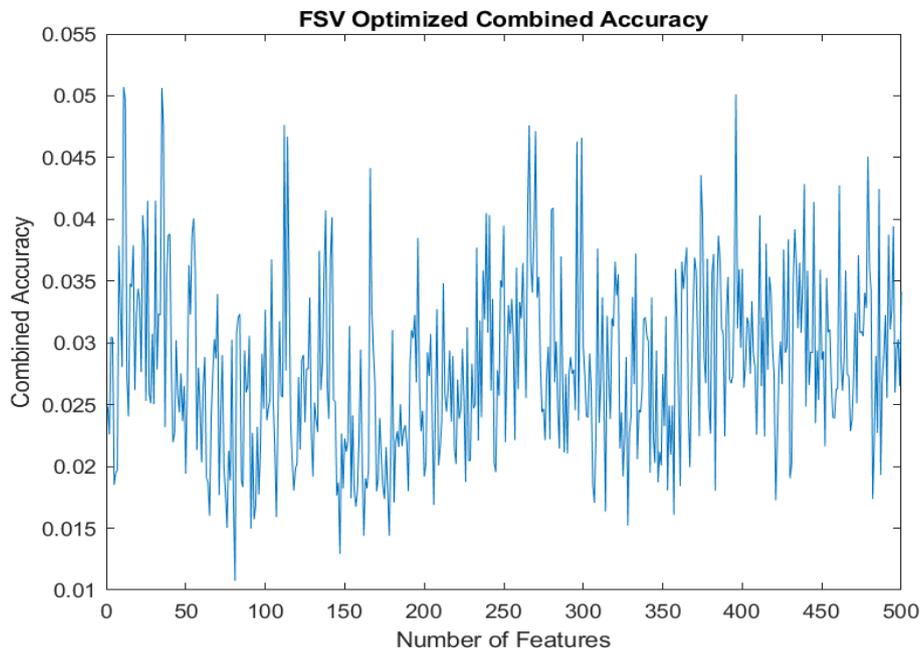


Figure 33: FSV Product of 5 Optimized classifiers accuracy

The FSV Optimized Combined Accuracy graph (fig 33) shows consistent but low performance, with combined accuracy fluctuating between 0.02 and 0.05 across the feature range. Despite the optimization process, the graph indicates no significant improvement in stability or peak combined accuracy compared to the non-optimized version. Peaks are sporadic and do not suggest a clear trend of improvement as more features are added. This suggests that the optimization process has limited impact on

enhancing the overall effectiveness of FSV, and the method struggles to select features that meaningfully improve classifier performance.

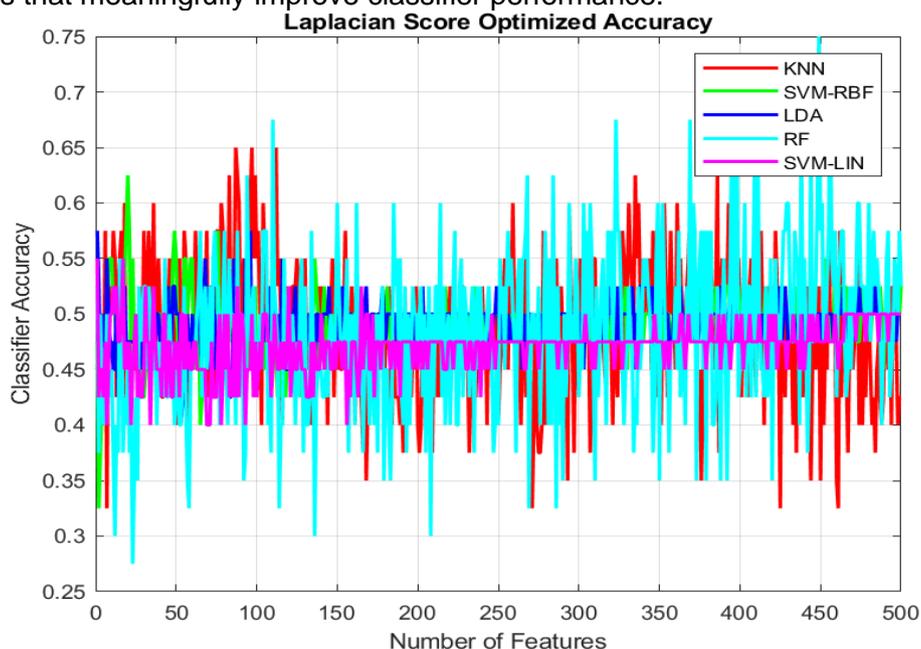


Figure 34: Laplacian Score Accuracy of 5 Optimized classifiers

The Laplacian Score Optimized Accuracy graph (fig 34) shows modest performance across classifiers, with accuracy generally stabilizing between 0.4 and 0.6 for most models. KNN and RF exhibit some variability with occasional spikes, while SVM-LIN remains consistently lower at around 0.4. LDA and SVM-RBF show relatively stable but moderate performance throughout. Compared to the non-optimized version, the optimization appears to have slightly reduced fluctuations, leading to more consistent accuracy, especially for RF and KNN. However, there is no significant overall improvement in accuracy, suggesting limited benefits of optimization for the Laplacian Score method. The results highlight that the Laplacian Score struggles to identify highly relevant features that can substantially boost classifier performance.

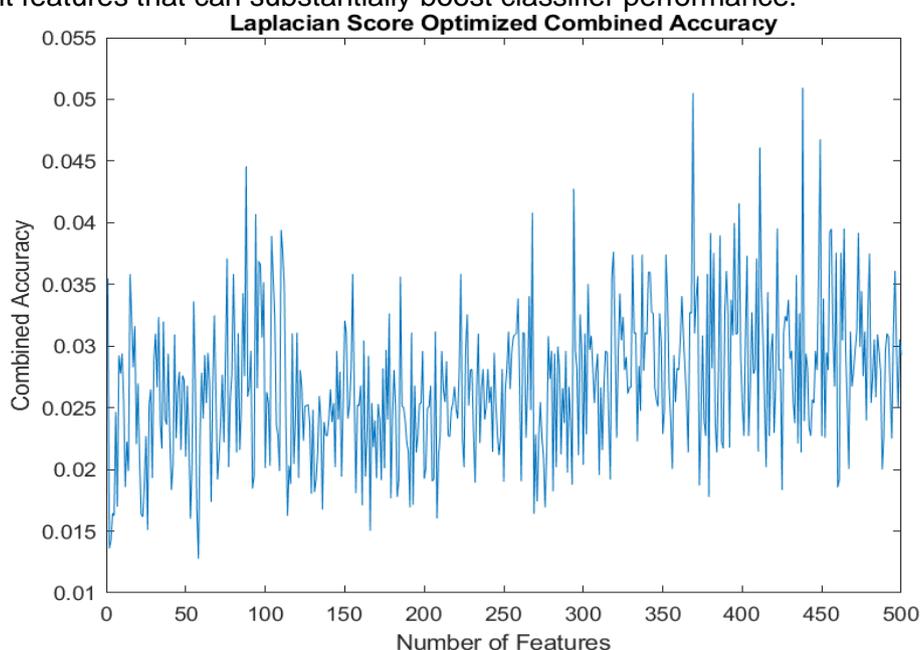


Figure 35: Laplacian Score Product of 5 Optimized classifiers accuracy

The Laplacian Score Optimized Combined Accuracy graph (fig 35) shows a fluctuating but consistently low performance, with combined accuracy oscillating between 0.02 and 0.045 across the feature range. Although the optimization process slightly stabilizes the fluctuations compared to the non-optimized version, it does not lead to

significant improvements in combined accuracy. This suggests that, even with optimization, the method struggles to provide meaningful boosts in overall classifier performance.

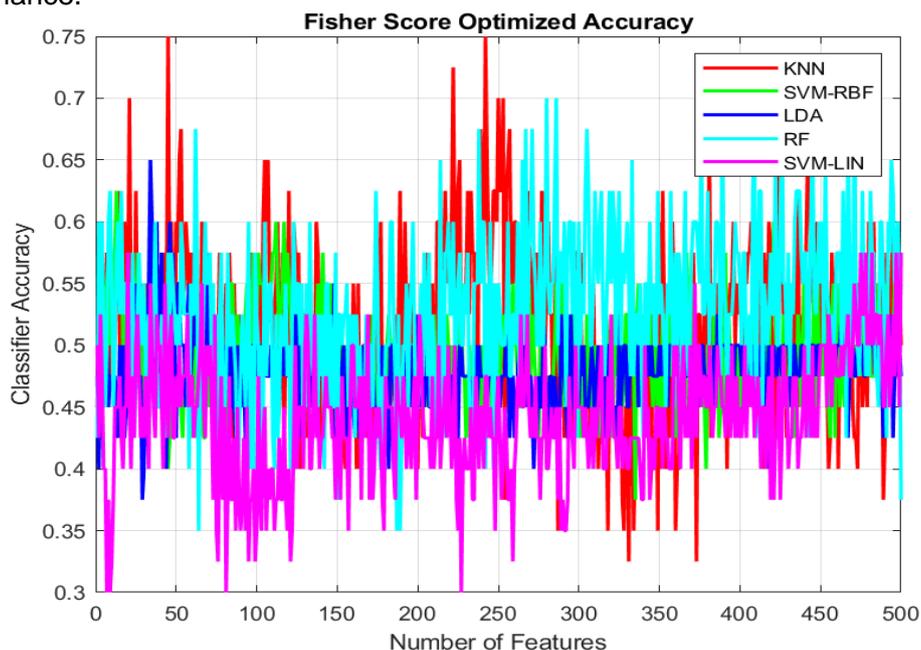


Figure 36: Fisher Score Accuracy of 5 Optimized classifiers

The Fisher Score Optimized Accuracy graph (fig 36) shows modest improvements in stability compared to the non-optimized version, but overall classifier performance remains moderate. The optimization process appears to reduce variability in RF and KNN's accuracy to some extent, but fluctuations persist across the range of features. While the optimization has slightly improved consistency, it has not significantly enhanced overall performance, indicating limited effectiveness in refining Fisher Score-selected features for meaningful accuracy gains.

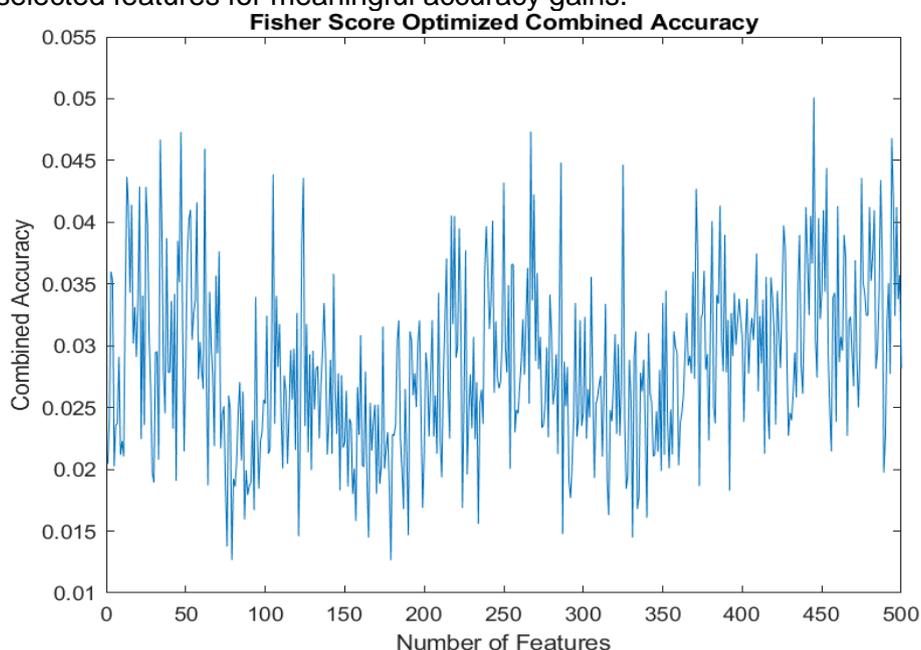


Figure 37: Fisher Score Product of 5 Optimized classifiers accuracy

The Fisher Score Optimized Combined Accuracy graph (fig 37) shows persistent fluctuations in combined accuracy, ranging between 0.02 and 0.045 across the feature range. Although the optimization process slightly reduces variability compared to the non-optimized version, it fails to produce any substantial improvement in combined accuracy. There is no clear trend of sustained improvement as more features are added. This indicates that while Fisher Score optimization may marginally enhance

stability, it struggles to significantly refine feature selection for boosting the overall performance of classifier ensembles.

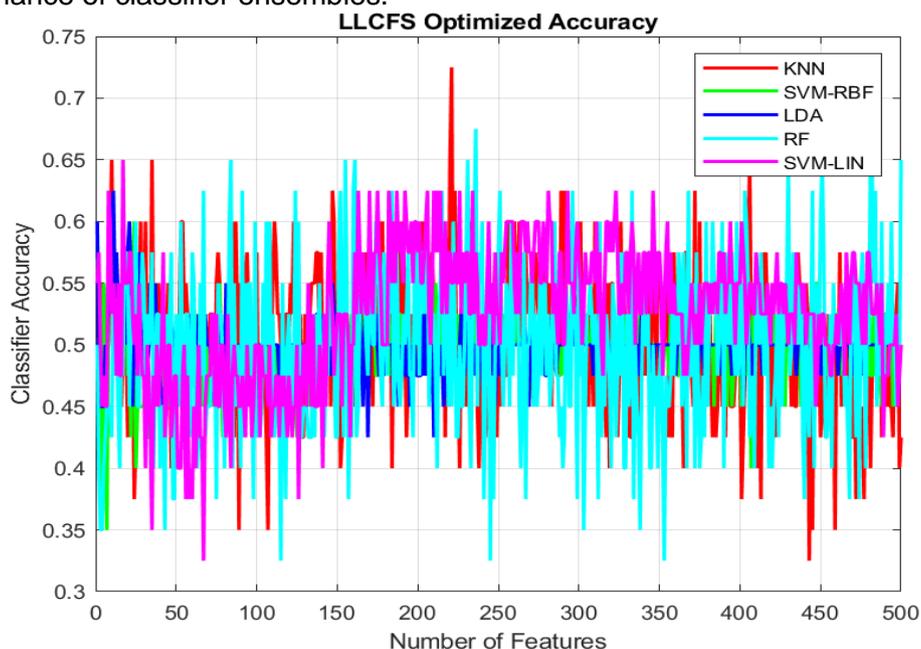


Figure 38: LLCFS Accuracy of 5 Optimized classifiers

The LLCFS Optimized Accuracy graph (fig 38) shows moderate performance across classifiers, with most accuracy values fluctuating between 0.45 and 0.6. Compared to the non-optimized version, the optimization slightly reduces variability but, the overall accuracy improvements are minimal, indicating that the optimization process does not significantly enhance LLCFS's ability to select highly relevant features for most classifiers. The results suggest limited effectiveness in improving classifier performance across the entire feature range.

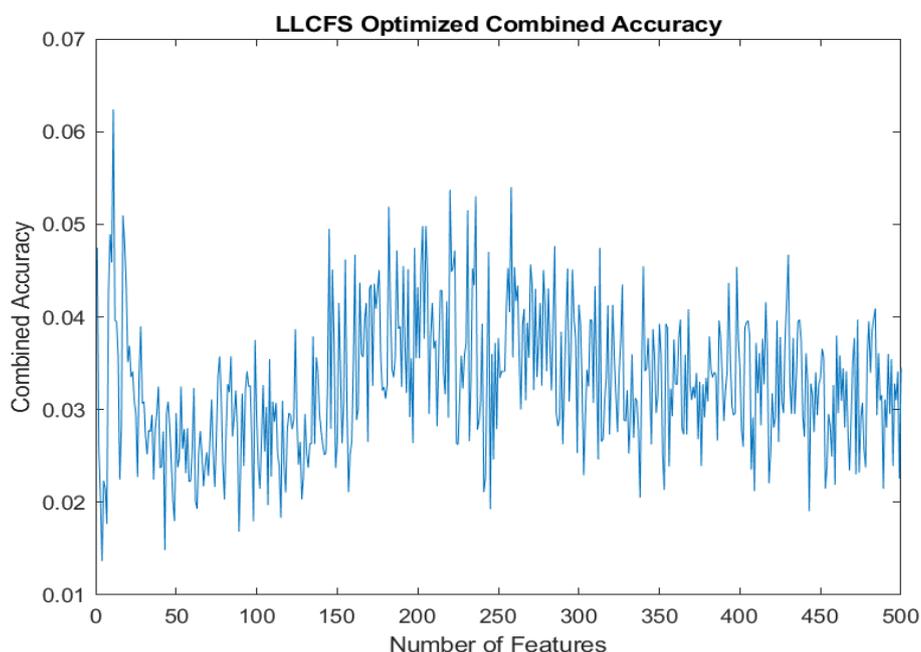


Figure 39: LLCFS Product of 5 Optimized classifiers accuracy

The LLCFS Optimized Combined Accuracy graph (fig 39) shows consistently low combined accuracy throughout the feature range. While optimization appears to reduce some variability, it does not result in a significant improvement in overall performance. These results indicate that the optimization of LLCFS provides limited

benefits, as it struggles to identify feature subsets that significantly enhance the overall performance of the classifiers.

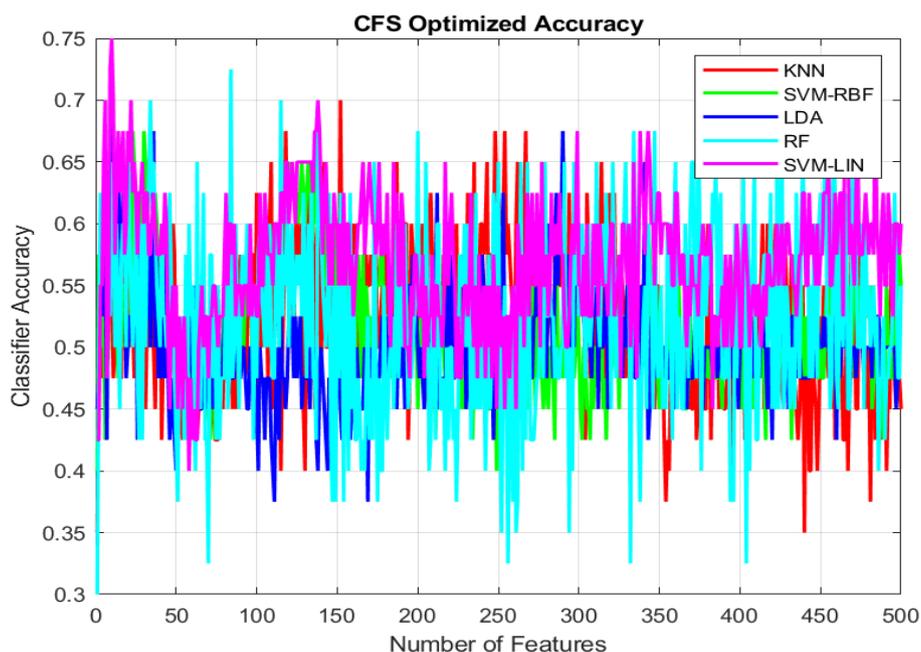


Figure 40: CFS Accuracy of 5 Optimized classifiers

The CFS Optimized Accuracy graph (fig 40) shows moderate performance across classifiers, with accuracy values fluctuating between 0.45 and 0.65 for most classifiers. The optimization process appears to slightly stabilize classifier accuracy, reducing variability for RF and KNN, but it does not result in substantial improvements in overall performance. These results suggest that while optimization enhances some aspects of CFS-selected features, it does not significantly improve their effectiveness for achieving consistently high classifier accuracy.

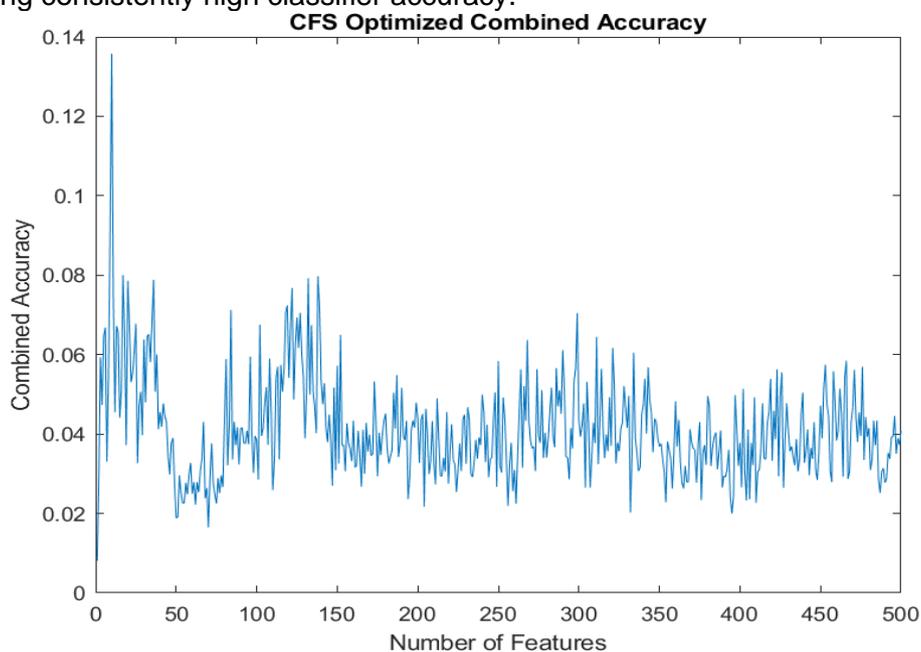


Figure 41: LLCFS Product of 5 Optimized classifiers accuracy

The CFS Optimized Combined Accuracy graph (fig 41) shows a sharp initial peak at around 0.12 for the first few features, followed by a rapid decline and stabilization around 0.04 for the majority of the feature range. While the optimization process may have slightly improved early feature selection, the combined accuracy remains low and fluctuates significantly as the number of features increases. Unlike the non-optimized version, the optimized version appears to stabilize earlier, but it does not provide a

substantial or sustained boost in combined accuracy. This suggests that the optimization of CFS has limited effectiveness in improving overall classifier ensemble performance across larger feature sets.

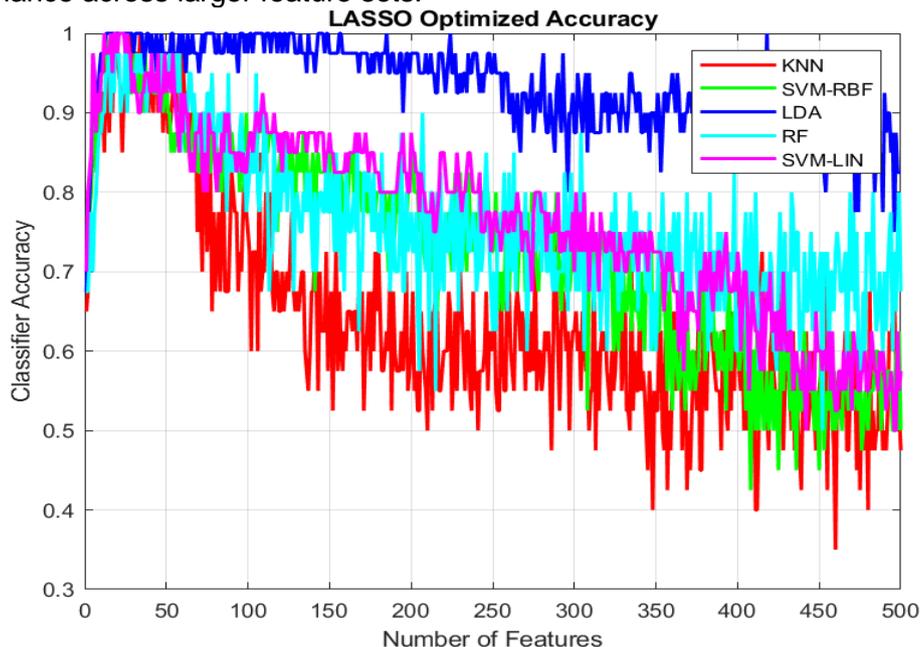


Figure 42: LASSO Accuracy of 5 Optimized classifiers

The LASSO Optimized Accuracy graph (fig 42) demonstrates exceptional performance, particularly for LDA, which maintains near-perfect accuracy across all feature ranges. The optimization process has clearly enhanced the performance of classifiers like, as their stability and accuracy remain high across increasing feature ranges. However, KNN's performance shows a noticeable downward trend as more features are included. Overall, LASSO optimization stands out for its ability to maintain strong classifier performance, making it a highly effective feature selection method.

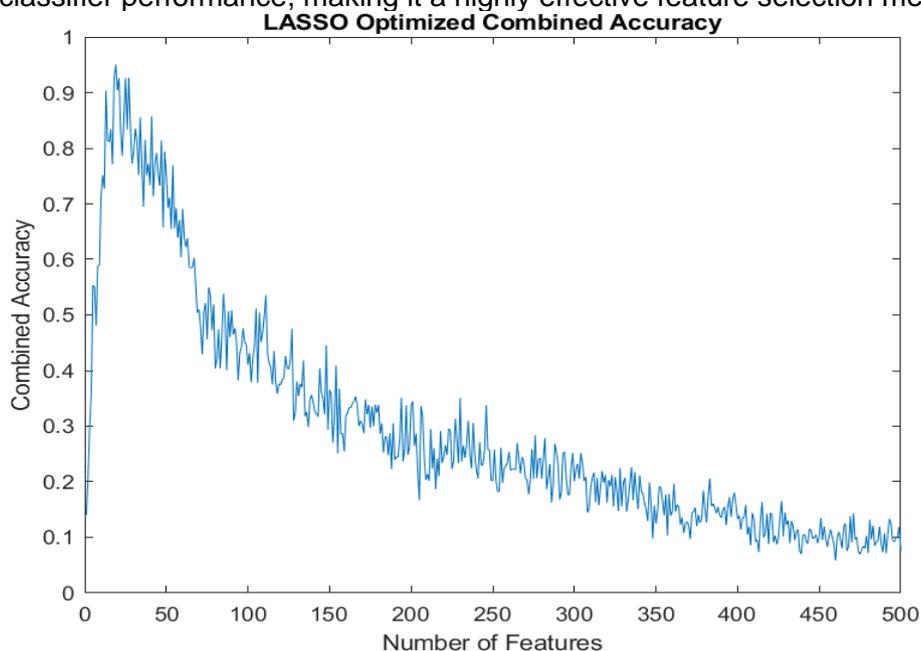


Figure 43: LASSO Product of 5 Optimized classifiers accuracy

The LASSO Optimized Combined Accuracy graph (fig 43) reveals that combined accuracy peaks very early, reaching approximately 0.95 with a small subset of features (around 20 features), and then gradually declines as more features are added. This indicates that LASSO optimization is particularly effective at selecting highly relevant features in the early stages, enabling excellent performance with minimal feature subsets. However, as the number of features increases, the combined accuracy diminishes. This suggests that while LASSO excels at feature selection for smaller

number of features, its performance diminishes with larger feature sets, likely due to the inclusion of less relevant features. Overall, the graph highlights LASSO's strength in early-stage feature optimization and suggests diminishing returns as more features are introduced.

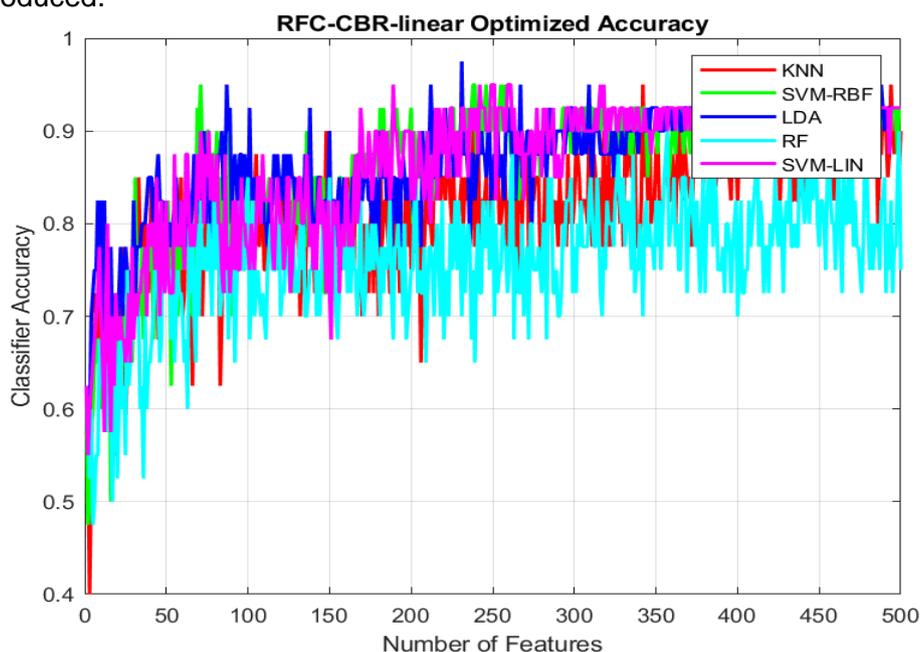


Figure 44: RFC-CBR-linear Accuracy of 5 Optimized classifiers

The RFC-CBR-linear Optimized Accuracy graph (fig 44) demonstrates consistently strong classifier performance across a wide range of features. Most classifiers, including LDA, RF, and SVM-RBF, achieve high accuracy after selecting approximately 100 features and maintain this performance throughout the range. KNN and SVM-LIN also exhibit steady improvement, with reduced variability as the number of features increases. The optimization process clearly enhances stability and scalability, allowing the method to maintain robust accuracy across classifiers. The results highlight the effectiveness of the RFC-CBR-linear optimization in providing consistently high classifier accuracy, making it a reliable feature selection method for large feature subsets.

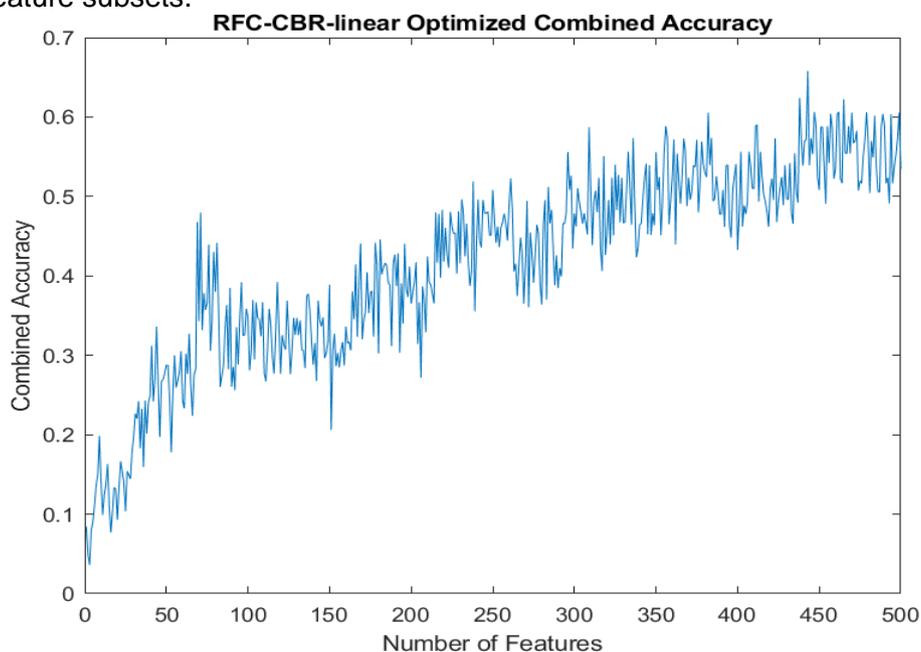


Figure 45: RFC-CBR-linear Product of 5 Optimized classifiers accuracy

The RFC-CBR-linear Optimized Combined Accuracy graph (fig 45) illustrates a steady and consistent improvement in combined accuracy as the number of features increases. Starting at around 0.1 with a small subset of features, the combined

accuracy rises steadily to reach approximately 0.6 at 500 features. This upward trend indicates that the optimization process effectively selects features that contribute meaningfully to overall classifier performance, with minimal noise or irrelevant features. The consistent improvement and stability across the feature range highlight the robustness and scalability of the RFC-CBR-linear optimization, making it a highly effective method for enhancing combined accuracy in ensemble classifiers.

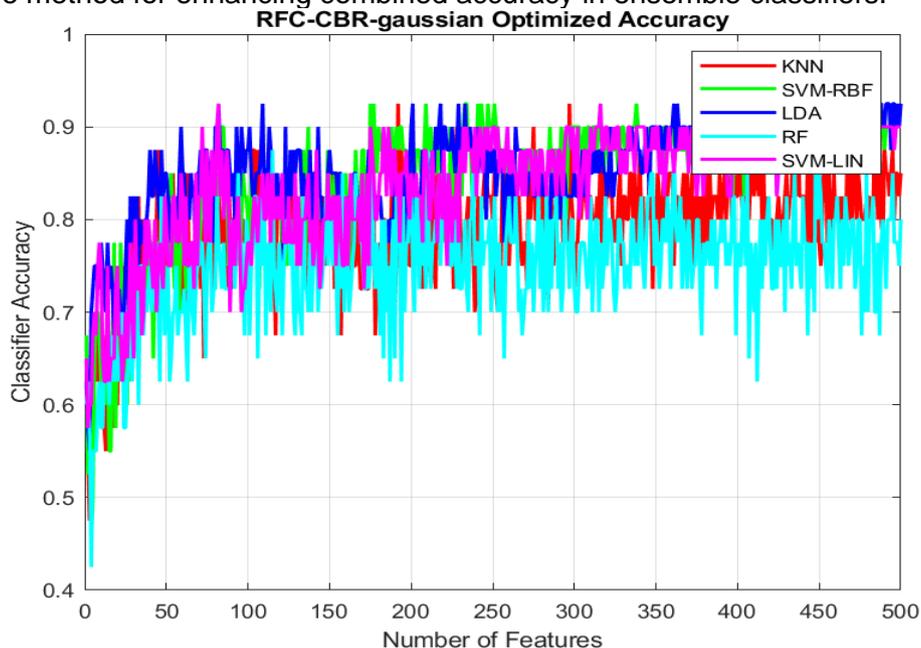


Figure 46: RFC-CBR-gaussian Accuracy of 5 Optimized classifiers

The RFC-CBR-gaussian Optimized Accuracy graph (fig 46) demonstrates strong and consistent classifier performance, with most classifiers achieving accuracy levels between 0.8 and 0.9 across the feature range. LDA and RF exhibit particularly stable and high accuracy throughout, while SVM-RBF and SVM-LIN also perform well, consistently reaching near 0.85. KNN shows slight variability but remains above 0.75 for most of the feature range. The optimization process effectively enhances accuracy across all classifiers, minimizing fluctuations and maintaining robust performance. These results highlight the strength of the RFC-CBR-gaussian method in identifying and utilizing relevant features, ensuring scalability and stability across the classifiers. The consistently high accuracy underscores the method's effectiveness in feature selection for improving overall classification performance.

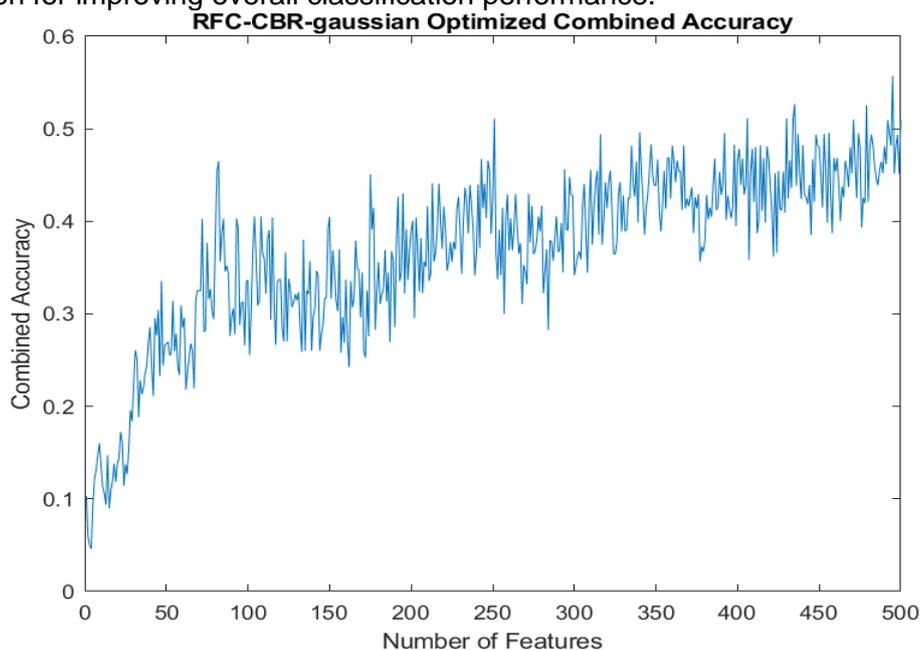


Figure 47: RFC-CBR-gaussian Product of 5 Optimized classifiers accuracy

The RFC-CBR-gaussian Optimized Combined Accuracy graph (fig 47) reveals a steady and consistent improvement in combined accuracy as the number of features increases. Starting at around 0.1 for smaller feature subsets, the combined accuracy rises steadily to approximately 0.5 as it approaches 500 features. This upward trend indicates that the optimization process effectively prioritizes relevant features, enhancing the overall performance of the ensemble classifiers. The smooth and gradual increase in combined accuracy, coupled with relatively low fluctuations, highlights the robustness of the RFC-CBR-gaussian optimization method. It effectively scales with larger feature sets, maintaining its capacity to improve classification accuracy consistently across the range.

The hyperparameters optimization indicates that **ReliefF**, **LASSO**, and the **RFC-CBR methods (linear and Gaussian)** were the most effective feature selection techniques, achieving significant improvements in classifier performance and stability. ReliefF excelled in identifying highly relevant features, enabling classifiers like KNN, RF, and LDA to achieve accuracies above 0.9 and demonstrating consistent, scalable performance as the feature set expanded. Similarly, LASSO showed exceptional results with small feature subsets, achieving near-perfect accuracy for LDA and a peak combined accuracy of 0.95 with around 20 features, making it the best method as fewer features are needed for successful classification. The RFC-CBR methods maintained robust and scalable performance across a wide range of features, with combined accuracies steadily increasing to 0.6 and 0.5 for the linear and Gaussian variants, respectively. In contrast, methods like ILFS, MutInfFS, and LLCFS struggled to provide meaningful accuracy improvements, even after optimization, reflecting their limited ability to prioritize highly relevant features.

3.3 Performance Metrics

In order to evaluate the feature selection algorithms, we will calculate the performance metrics for the number of features which provide the maximum combined classification accuracy as the product of all the classifiers [87]. In order to assess the performance of the Machine Learning algorithms included in this Thesis the classification accuracy, sensitivity and specificity were calculated with regard to the true vs the predicted labels.

- Accuracy is defined as the ratio of the number of correctly classified instances, i.e., the number of true positives plus the number of the true negatives, to the total number of instances.

$$Accuracy = \frac{\sum True\ Positives + \sum True\ Negatives}{\sum Total\ Number\ of\ cases}$$

- Sensitivity is the ratio of the number of true positives, to the total number of relevant positive elements.

$$Sensitivity = \frac{\sum True\ Positives}{\sum Total\ Number\ of\ Positives}$$

- Specificity is the ratio of the number of true negatives, to the total number of relevant negative elements

$$Specificity = \frac{\sum True\ Negatives}{\sum Total\ Number\ of\ Negatives}$$

- The F1-score is a metric used to evaluate the performance of a classification model. It is the harmonic mean of precision and recall, providing a single measure that balances both metrics.

$$F1Score = \frac{2TP}{2TP + FP + FN}$$

The maximum number of features we will perform the calculations will be 150 in order to ensure better explainability to our model and easier interpretation of the results

Table 2: ILFS Results

Feature selection algorithm		Best Combined Accuracy		Number Of Features	
ILFS		0.089		1	
Classifier	Accuracy	Sensitivity	Specificity	F1-Score	
KNN	0.725	0.7	0.75	0.75	
SVM-RBF	0.725	0.6	0.85	0.71	
LDA	0.5	0.95	0.05	0.67	
RF	0.75	1	1	1	
SVM-LIN	0.45	0.55	0.35	0.58	

Table 3: ReliefF Results

Feature selection algorithm		Best Combined Accuracy		Number Of Features	
ReliefF		0.753		132	
Classifier	Accuracy	Sensitivity	Specificity	F1-Score	
KNN	0.975	1	0.95	0.98	
SVM-RBF	0.925	0.95	0.9	0.93	
LDA	0.95	0.95	0.9	0.93	
RF	0.95	0.95	0.95	0.95	
SVM-LIN	0.925	0.95	0.9	0.93	

Table 4: MutLnFS Results

Feature selection algorithm		Best Combined Accuracy		Number Of Features	
MutLnFS		0.062		25	
Classifier	Accuracy	Sensitivity	Specificity	F1-Score	
KNN	0.575	0.5	0.65	0.6	
SVM-RBF	0.65	0.75	0.55	0.72	
LDA	0.525	0.55	0.5	0.6	
RF	0.55	0.6	0.5	0.63	
SVM-LIN	0.575	0.55	0.6	0.62	

Table 5: FSV Results

Feature selection algorithm		Best Combined Accuracy		Number Of Features	
FSV		0.041		11	
Classifier	Accuracy	Sensitivity	Specificity	F1-Score	
KNN	0.625	0.65	0.6	0.68	
SVM-RBF	0.5	0.2	0.8	0.32	
LDA	0.525	0.7	0.35	0.65	
RF	0.625	0.5	0.75	0.62	
SVM-LIN	0.4	0.35	0.45	0.45	

Table 6: Laplacian Score Results

Feature selection algorithm		Best Combined Accuracy		Number Of Features	
Laplacian Score		0.032		88	
Classifier	Accuracy	Sensitivity	Specificity	F1-Score	
KNN	0.6	0.6	0.6	0.65	
SVM-RBF	0.5	0.85	0.15	0.66	
LDA	0.5	0.75	0.250	0.65	
RF	0.4	0.45	0.35	0.52	
SVM-LIN	0.525	0.7	0.35	0.65	

Table 7: Fisher Score Results

Feature selection algorithm		Best Combined Accuracy		Number Of Features	
Fisher Score		0.043		47	
Classifier	Accuracy	Sensitivity	Specificity	F1-Score	
KNN	0.6	0.65	0.55	0.67	
SVM-RBF	0.475	0.45	0.5	0.54	
LDA	0.5	0.75	0.25	0.65	
RF	0.55	0.7	0.4	0.66	
SVM-LIN	0.55	0.45	0.65	0.56	

Table 8: LLCFS Results

Feature selection algorithm		Best Combined Accuracy		Number Of Features	
LLCFS		0.057		11	
Classifier	Accuracy	Sensitivity	Specificity	F1-Score	
KNN	0.6	0.6	0.6	0.65	
SVM-RBF	0.575	0.7	0.45	0.67	
LDA	0.6	0.65	0.55	0.67	
RF	0.525	0.5	0.55	0.58	
SVM-LIN	0.525	0.4	0.65	0.52	

Table 9: CFS Results

Feature selection algorithm		Best Combined Accuracy		Number Of Features	
CFS		0.13		10	
Classifier	Accuracy	Sensitivity	Specificity	F1-Score	
KNN	0.55	0.75	0.35	0.67	
SVM-RBF	0.675	0.6	0.75	0.69	
LDA	0.75	0.8	0.7	0.78	
RF	0.625	0.7	0.55	0.69	
SVM-LIN	0.75	0.85	0.65	0.79	

Table 10: LASSO Results

Feature selection algorithm		Best Combined Accuracy		Number Of Features	
LASSO		0.975		19	
Classifier	Accuracy	Sensitivity	Specificity	F1-Score	
KNN	0.975	1	0.95	0.98	
SVM-RBF	1	1	1	1	
LDA	1	1	1	1	
RF	1	1	1	1	
SVM-LIN	1	1	1	1	

Table 11: RFB-CBR-Linear Results

Feature selection algorithm		Best Combined Accuracy		Number Of Features	
RFB-CBR-Linear		0.402		71	
Classifier	Accuracy	Sensitivity	Specificity	F1-Score	
KNN	0.75	0.8	0.7	0.78	
SVM-RBF	0.9	0.95	0.85	0.9	
LDA	0.8	0.9	0.7	0.83	
RF	0.85	0.85	0.85	0.86	
SVM-LIN	0.875	0.9	0.85	0.88	

Table 12: RFB-CBR-Gaussian Results

Feature selection algorithm		Best Combined Accuracy		Number Of Features	
RFB-CBR-Gaussian		0.4		82	
Classifier	Accuracy	Sensitivity	Specificity	F1-Score	
KNN	0.7	0.8	0.6	0.75	
SVM-RBF	0.825	0.95	0.7	0.85	
LDA	0.9	0.95	0.85	0.9	
RF	0.875	0.9	0.85	0.88	
SVM-LIN	0.875	0.9	0.85	0.88	

The top four feature selection algorithms—LASSO, ReliefF, RFB-CBR-Linear, and RFB-CBR-Gaussian—are distinguished by their combined accuracy and classifier performance, as indicated by the calculated performance metrics.

LASSO demonstrated the highest combined accuracy of 0.975 with only 19 features, achieving perfect performance across all classifiers and metrics (Accuracy, Sensitivity, Specificity, and F1-Score). This result highlights its ability to extract the most relevant features while maintaining model simplicity and interpretability. LASSO's exceptional balance of accuracy and minimal feature count makes it the most effective algorithm for this dataset.

ReliefF achieved a combined accuracy of 0.753 with 132 features, demonstrating robust classification performance across multiple classifiers. It yielded particularly high accuracy for KNN (0.97) and SVM-RBF (0.925), though the large number of features limits its interpretability compared to LASSO.

RFB-CBR-Linear achieved a combined accuracy of 0.402 with 71 features, offering a solid balance between feature count and classifier performance. Classifiers such as SVM-RBF and LDA performed particularly well, with accuracy values of 0.9 and 0.8, respectively.

RFB-CBR-Gaussian demonstrated a similar combined accuracy of 0.4 with 82 features. While its accuracy is slightly lower than RFB-CBR-Linear, it still exhibited strong performance across classifiers like SVM-RBF and LDA, with high accuracy (0.825 and 0.9, respectively). This method complements RFB-CBR-Linear and is suitable for applications with a higher tolerance for feature dimensionality.

Overall, LASSO stands out as the most effective feature selection algorithm for this dataset, achieving an outstanding combined accuracy of 0.975 while utilizing only 19 features. Its ability to select a minimal yet highly informative subset of features ensures not only exceptional classification performance but also enhanced interpretability of the model. LASSO's adaptability and robustness to a variety of machine learning models are demonstrated by its acquisition of ideal metrics across all classifiers—KNN, SVM-RBF, LDA, Random Forest, and SVM-LIN. Specifically, it achieves 100% sensitivity and specificity across all classifiers, indicating its ability to accurately identify both rest and fatigue states without any false negatives or false positives. Moreover, the F1-score of 1.0 across all classifiers highlights its strong balance between precision and recall, further validating its reliability in capturing the true patterns in the data.

The minimal feature count selected by LASSO significantly reduces model complexity, making it easier to interpret and apply in practical scenarios, such as clinical or real-time applications where computational efficiency and clarity are essential. Its performance across diverse classifiers also suggests that the selected features generalize well to different machine learning techniques, reinforcing its utility as a robust feature selection method.

3.4 Functional Connectivity Features

In the previous chapter we calculated that **LASSO** demonstrated the highest combined accuracy of 0.975 with only 19 features, achieving perfect performance across all classifiers. We will examine those 19 features of the PLI matrix in order to extract the qualitative characteristics of the connections using a MATLAB® algorithm that translates the those features to Brean network connections.

This algorithm identifies and interprets key EEG channels and their connectivity based on selected features derived from a ranking matrix created from LASSO. The algorithm focuses on the top 19 features, which are chosen and ranked based on their frequency of occurrence across multiple cross-validation folds. These features represent connectivity patterns in the brain, extracted from the upper triangular part of a 63x63 Phase Lag Index (PLI) matrix. The PLI matrix quantifies the connectivity between EEG channels by examining phase synchronization.

Each feature is mapped to a specific connection between two EEG channels and is associated with one of the five frequency bands of brain waves: Delta (1–4 Hz), Theta (4–7 Hz), Alpha (8–12 Hz), Beta (13–30 Hz), and Gamma (31–45 Hz). By determining the index of each feature, the algorithm calculates the frequency band it belongs to and identifies the relative position within that band. Using a predefined lookup table, the algorithm converts the 1D index of the feature into row and column indices of the PLI matrix, which correspond to specific EEG channel pairs.

The predefined list of 63 EEG channels is then used to map these indices to their respective channel names, creating a detailed list of channel connections and their associated frequency bands. The final output is stored in a cell array that includes the source channel, the target channel, and the frequency band for each identified connection.

This output provides insights into the functional connectivity of the brain, allowing the exploration of the relationships between different regions and how these connections vary across frequency bands.

Table 13: Most Significant Features Identified By LASSO

EEG Channel	'C4'	'CP1'	'FC4'	'C1'	'Pz'	'F6'	'POz'
EEG Channel	'Cz'	'F1'	'Fp2'	'FC5'	'C1'	'Fp2'	'TP7'
Frequency Band	'Beta'	'Delta'	'Theta'	'Gamma'	'Theta'	'Theta'	'Gamma'
EEG Channel	'FC6'	'P6'	'F6'	'C5'	'CP6'	'P4'	'PO7'
EEG Channel	'F4'	'C1'	'F4'	'FC1'	'CP5'	'AF8'	'AF7'
Frequency Band	'Delta'	'Theta'	'Delta'	'Delta'	'Delta'	'Delta'	'Alpha'
EEG Channel	'Fz'	'P1'	'C2'	'F4'	'AF4'		
EEG Channel	'F1'	'C3'	'F6'	'Fpz'	'F5'		
Frequency Band	'Theta'	'Theta'	'Theta'	'Alpha'	'Beta'		

3.4.1 Brain Wave band analysis

The first analysis of channel connections will be their associated frequency bands of the brain waves. Brain waves, characterized by their distinct frequency ranges, play crucial roles in various neural processes and are indicative of different cognitive and physiological states. By examining the connections within these frequency bands, we can gain valuable insights into how neural activity differs between rest and fatigue conditions.

Table 14: Distribution of connections in frequency bands

Frequency Band	Connection Counts
Delta (1–4 Hz)	6
Theta (4–7 Hz)	7
Alpha (8–12 Hz)	2
Beta (13–30 Hz)	2
Gamma (31–45 Hz)	2

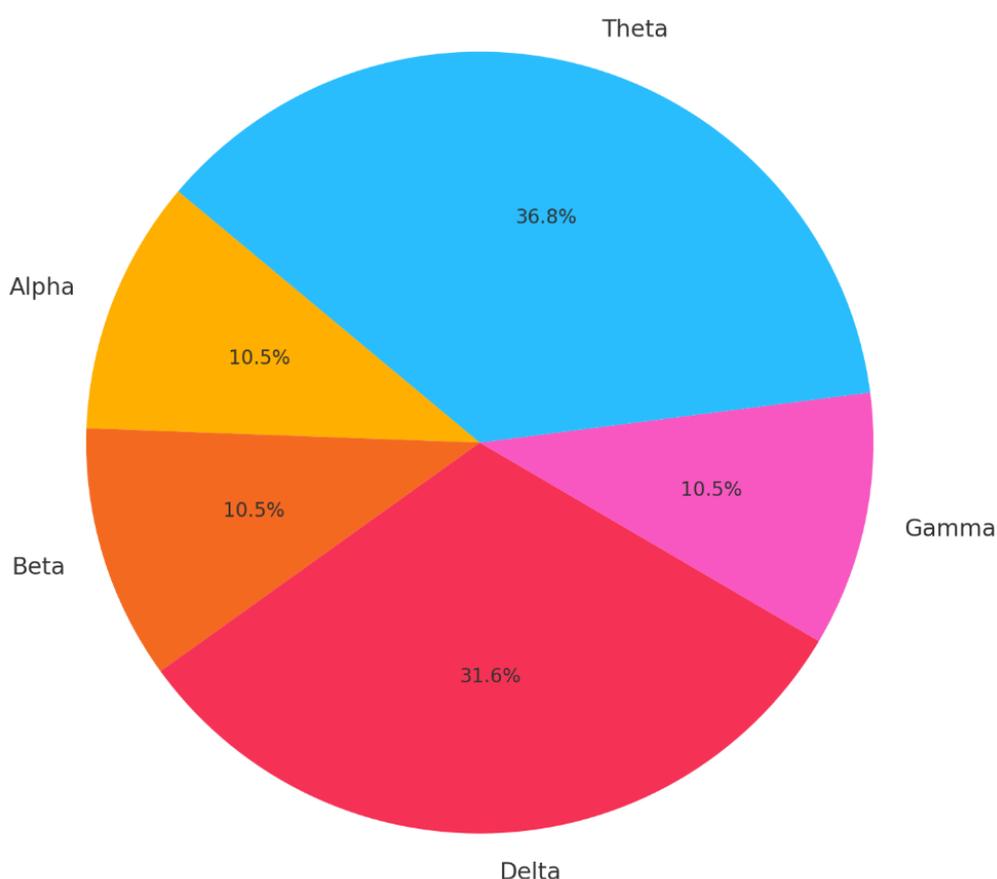


Figure 48: Frequency Band Connection Distribution

The pie chart (Figure 48) highlights the distribution of PLI connections across the five primary EEG frequency bands—Delta, Theta, Alpha, Beta, and Gamma—emphasizing their relative contributions to differentiating between rest and fatigue states.

The **Theta band (4–7 Hz)** indicated with blue color is the most dominant, representing 36.8% of the total connections. Theta activity is widely recognized for its association with cognitive effort, sustained attention, and fatigue. During prolonged tasks requiring mental focus, the brain often exhibits increased theta activity, reflecting a compensatory mechanism to manage reduced cognitive efficiency. This band plays

a critical role in monitoring mental states, as its heightened connectivity in fatigue states is indicative of the brain's efforts to sustain performance despite the adverse effects of fatigue. The prominence of theta oscillations underscores their significance as a marker for fatigue, often linked to frontal and central regions where cognitive processing and attentional control are most active [88], [89], [90].

The **Delta band (1–4 Hz)**, indicated with red color, accounting for 31.6% of the connections, is the second most significant contributor. Delta activity is primarily associated with deep cortical recovery, slow-wave sleep, and restorative processes. Its increased connectivity during fatigue states likely reflects the brain's need for recovery and the engagement of slower oscillatory activity as a response to reduced neural efficiency. Delta activity is also implicated in large-scale network synchronization, which may become more pronounced during fatigue as the brain attempts to preserve homeostasis and manage the impact of prolonged exertion. The high representation of delta oscillations in the data suggests their critical involvement in both physical and mental fatigue, as these oscillations often manifest in states requiring recovery or rest [91], [92], [93], [94], [95].

The **Alpha band (8–12 Hz)** indicated with yellow color, contributes 10.5% of the connections, playing a smaller role in distinguishing between rest and fatigue states. Alpha activity is typically linked to relaxed wakefulness and cortical inhibition, reflecting states of reduced sensory input and increased focus on internal processing. Variations in alpha connectivity between rest and fatigue states may represent shifts in cortical balance, with the brain transitioning between externally and internally focused states.[33], [40], [43] The smaller contribution of alpha connectivity suggests it is secondary to the more pronounced roles of theta and delta bands.

Similarly, the **Beta band (13–30 Hz)**, indicated with orange color, accounts for 10.5% of the connections. Beta activity is associated with motor and cognitive functions, and its involvement in this dataset may reflect increased neural effort or stress-related processes during fatigue. When the brain is fatigued, maintaining performance often requires greater motor readiness and cognitive activation, which may explain the role of beta oscillations[54], [55].

The **Gamma band (31–45 Hz)**, indicated with purple color, also contributing 10.5% of the connections, is associated with higher-order cognitive functions and localized neural activity. While its contribution is less prominent than the theta and delta bands, gamma connectivity may play a role in modulating fatigue-related changes in cognitive processing and task-specific neural dynamics. Gamma activity is often linked to neural synchronization in processes like working memory and attention, which may become relevant during fatigue [96], [97], [98].

Overall, the **Theta and Delta bands** dominate the observed connectivity patterns, reflecting their essential roles in neural processes impacted by fatigue. Theta connectivity is particularly critical in compensating for reduced cognitive efficiency, highlighting its role in sustaining attention and cognitive control. Delta connectivity, on the other hand, emphasizes the brain's restorative efforts and large-scale synchronization in response to fatigue. This distribution aligns with the understanding that lower-frequency bands are most influenced during states of fatigue, with higher-frequency bands contributing to more localized or specific neural responses.

3.4.2 Channel Connection Analysis

The second analysis of channel connections will be their associated regions of the brain which have increased connectivity. In order to examine the topological characteristics of the connections we review the electrode locations of 64 EEG channel according to the International 10-20 system.

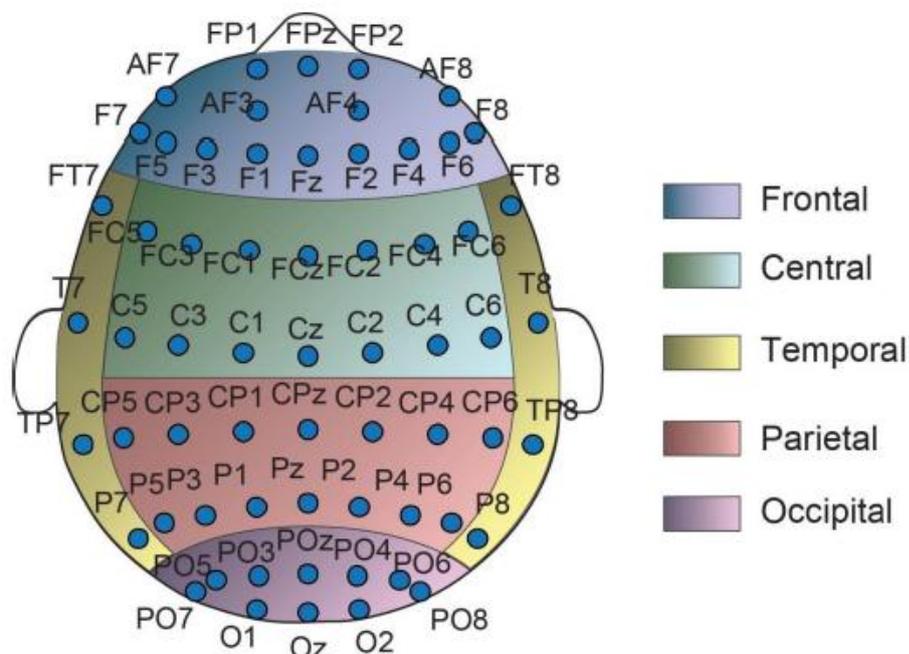


Figure 49: Electrode locations of 64 EEG channel

The EEG channels which are activated and indicated by the PLI connections are distributed topographically by brain region. The brain is divided into five main lobes based on anatomical landmarks and functional specializations. Each lobe is associated with specific functions and is delineated by sulci (grooves) and gyri (ridges) on the brain's surface.

Table 15: Topology of connections

Brain Region	Channels in PLI connections
Frontal Lobe	16
Central Lobe	12
Parietal Lobe	7
Occipital Lobe	1
Temporal Lobe	2

Distribution of EEG Channels in PLI Connections by Brain Region

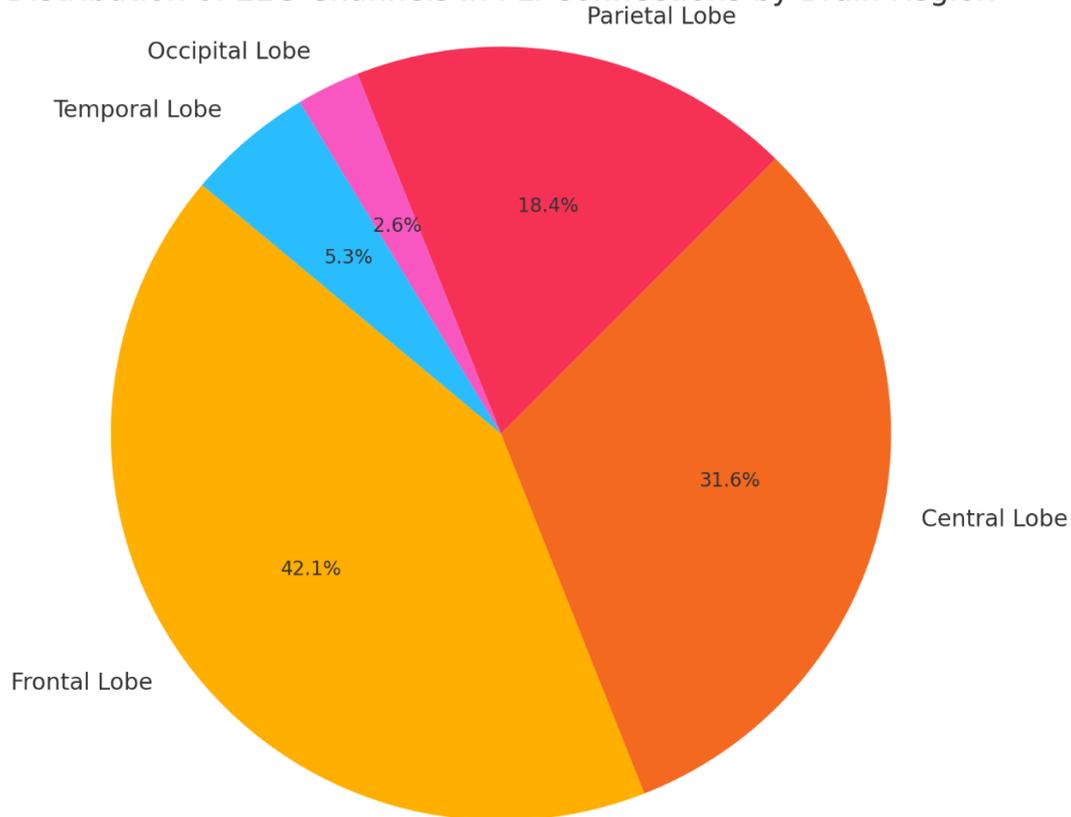


Figure 50: Distribution of EEG channel connections by Brain region

The distribution (Figure 50) indicates that the connectivity patterns documented by the PLI method are primarily concentrated in the **Frontal** and **Central** Lobes, underscoring their critical roles in the neural processes under investigation. While the **Temporal** and **Occipital** Lobes exhibit limited contributions, the **Parietal** Lobe contributes moderately, suggesting that task-specific or methodological factors may be influencing their involvement.

The **Frontal Lobe**, which accounts for **42.1%** of the connections (yellow color Fig 50), is the most significant region in this analysis, as it plays a critical role in executive functions, decision-making, and attentional control. The brain's capacity to govern attention, manage complex tasks, and sustain efficient decision-making is frequently impaired by fatigue, which increases the vulnerability of these processes, which are central to cognitive performance [36], [37]. The frontal lobe's substantial connectivity underscores its significance in mitigating these effects, indicating that the brain reallocates resources within this region to maintain cognitive function during exhaustion. The increased connectivity of the frontal lobe during fatigue is likely a result of compensatory mechanisms that are designed to maintain task performance in the presence of strained cognitive resources. This increased activity is indicative of the brain's attempt to mitigate the cognitive delay that is typically associated with exhaustion, thereby guaranteeing that critical functions, including adaptive responses and attentional control, continue to function [33], [43], [52].

This adaptation is essential for maintaining performance in tasks that require sustained focus or significant mental effort, where the frontal lobe's ability to manage executive control becomes indispensable. The analysis's emphasis on the frontal lobe emphasizes its function as a central center for cognitive processing, coordinating neural responses to address the obstacles presented by fatigue. The frontal lobe facilitates the brain's capacity to adapt dynamically by improving connectivity, thereby prioritizing functions that are essential for task execution and compensating for reduced cognitive efficacy. This adaptability underscores the region's importance in

promoting resilience against fatigue-related impairments and its indispensable role in the maintenance of cognitive performance [94], [95], [99], [100].

The **Central Lobe**, which accounts for **31.6%** of the connections (orange color Fig 50), is a critical region in the differentiation between rest and fatigue states, as it plays a fundamental role in the integration of motor and sensory information. This region is responsible for the mediation of critical functions, including sensory perception, reaction timing, and movement coordination. The prevalent impact of fatigue on sensorimotor processing is underscored by the substantial representation of central lobe connectivity in the analysis. Fatigue frequently manifests as impaired sensory responsiveness, reduced motor precision, and delayed reaction times, all of which are functions that are mediated by the central lobe [101], [102]. For example, the central lobe's diminished ability to coordinate fine motor activities during fatigue can lead to reduced motor precision, while sensory responsiveness changes suggest that external stimuli are not being processed effectively. The central lobe's substantial connectivity during fatigue implies that this region adjusts by reallocating neural resources to account for the decrease in energy levels. The objective of these adaptive mechanisms is to maintain sensory processing and motor readiness in the face of the cognitive and physiological strain that fatigue induces. This dynamic adjustment is indicative of the brain's overarching strategy to efficiently manage energy while maintaining critical functions such as perception and movement [103], [104]. These findings underscore the central lobe's critical contribution to maintaining functional integrity and adapting to the challenges posed by prolonged cognitive or physical exertion.

The **Parietal Lobe**, accounting for **18.4%** of the connections (red color Fig 50), plays a significant role in spatial awareness, sensory integration, and attentional modulation. Fatigue can disrupt these processes, resulting in a decreased capacity to process sensory information effectively and a decrease in spatial accuracy. The parietal lobe's moderate contribution to the analysis implies that it offers critical insights into the neural adjustments that occur in response to fatigue. The involvement of this region may be indicative of modifications in the processing and integration of sensory and spatial information, which are essential for the preservation of task performance in challenging environments [88], [90], [105].

Limited involvement in this analysis is demonstrated by the **Temporal Lobe**, (blue color Fig 50) which accounts for only **5.3%** of the connections. The temporal lobe's diminished contribution may suggest that fatigue has a less significant impact on these processes in the context of this dataset or task, as it is primarily associated with auditory processing and memory functions. The limited connectivity implies that auditory and memory-related functions are relatively stable across rest and fatigue states, or that their differentiation is less critical for this analysis.

Among all cerebral regions, the **Occipital Lobe** (purple color Fig 50) makes the least contribution, accounting for only **2.6%** of the connections. In this investigation, visual processing, which is regulated by the occipital lobe, is not a primary differentiator between rest and fatigue. This finding is consistent with this understanding. The occipital lobe's minimal involvement in distinguishing between these states is likely due to the fact that the task or dataset does not significantly involve visual stimuli.

The Frontal and Central Lobes are the primary regions involved in the differentiation between rest and fatigue states, as the findings reveal. These regions underscore the cognitive and sensorimotor disruptions that are indicative of fatigue, such as impaired executive functions, attentional control, and motor readiness. The Parietal Lobe also makes a significant contribution, as it reflects alterations in spatial processing and sensory integration. These results offer a comprehensive understanding of the neural dynamics that are involved in rest and fatigue, providing valuable insights into the specific brain regions and connections that are most affected.

3.4.3 Topological Connectivity Analysis

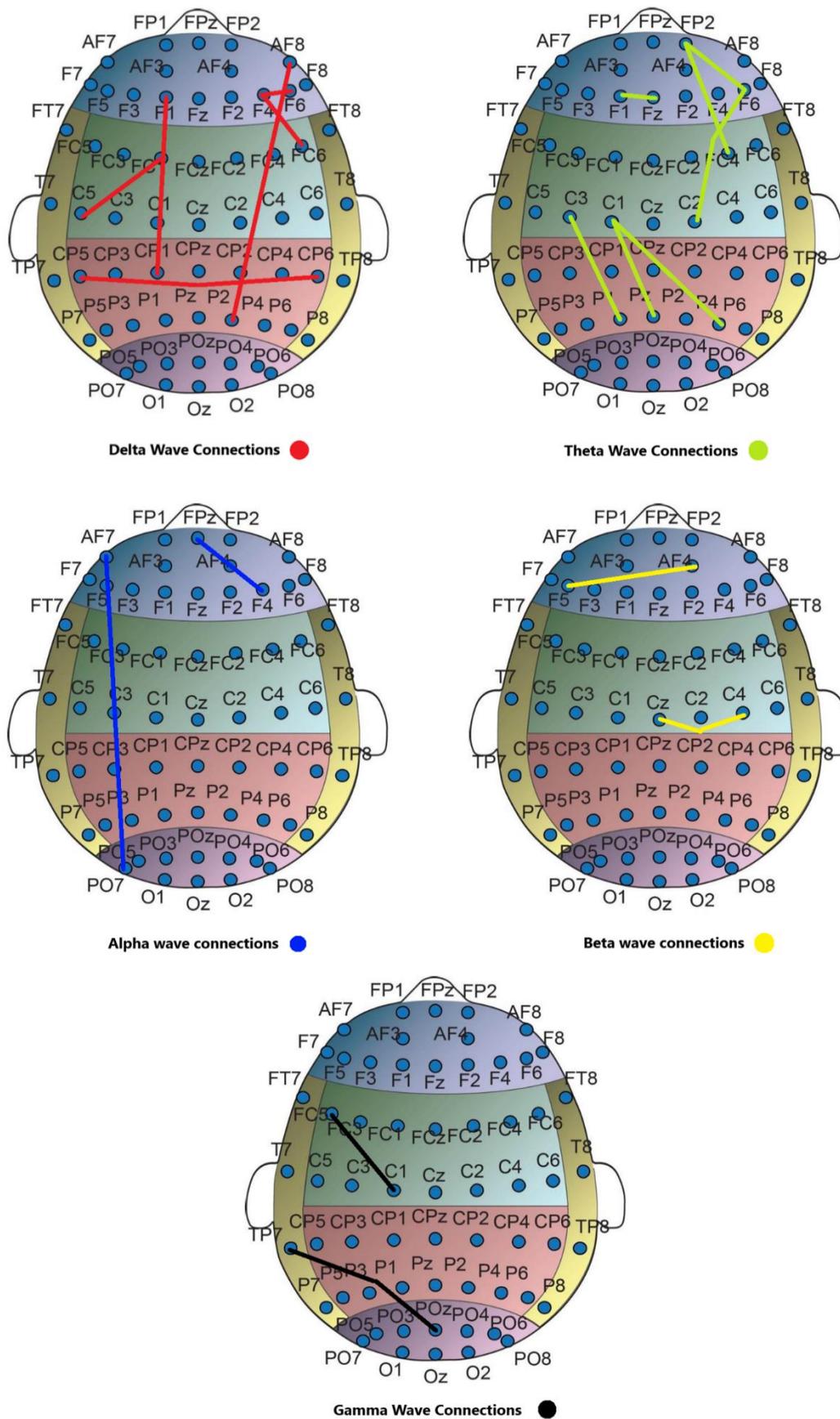


Figure 51: Brain wave connections across different brain lobes

Table 16: Topology of brain wave connections

Brain Wave rhythm	Frontal Lobe	Central Lobe	Parietal Lobe	Temporal Lobe	Occipital Lobe
Delta	4	3	4	0	0
Theta	4	4	3	0	0
Alpha	3	0	0	0	1
Beta	2	2	0	0	0
Gamma	0	2	0	1	1

The results presented in figure 51 highlight the brain wave connections (Delta, Theta, Alpha, Beta, Gamma) across different brain lobes (Frontal, Central, Parietal, Temporal, and Occipital) that demonstrated the maximum differentiation between rest and fatigue states. These findings reflect the neural dynamics underlying the transition between these states and emphasize the critical role of specific brain rhythms in distinguishing between rest and fatigue conditions:

- **Delta Wave Connections:** Delta waves show a broad distribution across the Frontal (4 connections), Central (3 connections), and Parietal lobes (4 connections). These slow-wave rhythms, typically associated with restorative processes and deep sleep, likely capture the brain's response to the need for recovery during fatigue. The high differentiation suggests that Delta activity may reflect fatigue-induced changes in neural synchrony, particularly in sensory and motor integration regions.
- **Theta Wave Connections:** Theta rhythms exhibit balanced connectivity across the Frontal (4 connections), Central (4 connections), and Parietal lobes (3 connections). Theta waves are widely recognized for their role in memory consolidation and attentional regulation. Their significant differentiation between rest and fatigue states suggests that Theta activity may be modulated to sustain cognitive and attentional demands during fatigue, particularly in regions involved in sensory processing and higher-order cognition.
- **Alpha Wave Connections:** Alpha rhythms display fewer connections, predominantly in the Frontal lobe (3 connections) and a single connection in the Occipital lobe. These rhythms, associated with relaxation and sensory inhibition, may play a more localized role in distinguishing rest and fatigue. The reduced connectivity suggests that Alpha waves are primarily involved in modulating sensory processing (evident in the Occipital lobe) and maintaining cognitive balance in the Frontal regions, which are critical for decision-making and attentional regulation.
- **Beta Wave Connections:** Beta rhythms, with limited connectivity in the Frontal (2 connections) and Central lobes (2 connections), reflect their specialized role in supporting active thinking, motor coordination, and decision-making. The fewer connections emphasize Beta waves' targeted involvement in processes strained by fatigue, such as motor readiness and task-relevant cognitive functions, rather than a broader neural network response.
- **Gamma Wave Connections:** Gamma rhythms, the fastest brain waves, show the least connectivity, with differentiation observed in the Central (2 connections), Temporal (1 connection), and Occipital lobes (1 connection). Associated with high-level cognitive integration, Gamma waves likely represent localized neural activity changes in response to fatigue. Their minimal but specific connections suggest their role in maintaining integrative processes, such as sensory coordination and temporal resolution, which are crucial in high-demand cognitive states.

The distribution of these rhythms highlights how different neural mechanisms adapt to distinguish rest and fatigue states. Delta and Theta waves, with their extensive connectivity, reflect broad neural adaptations in response to fatigue, particularly in restorative and attentional processes. In contrast, Alpha, Beta, and Gamma rhythms,

with fewer connections, likely represent specialized roles in sensory processing, cognitive integration, and motor control. These findings underscore the complexity of brain rhythms in regulating neural dynamics and provide valuable insights for understanding and monitoring fatigue-related neural activity

4 Discussion

This thesis combined sophisticated feature selection techniques and machine learning algorithms to provide a thorough framework for the classification of rest and fatigue states using EEG data. The methodology employed in this study was grounded in a multi-stage analysis pipeline. EEG data from 20 participants were recorded following the 10-20 electrode placement standard. Raw EEG data underwent necessary preprocessing processes including bandpass filtering and artifact removal to guarantee high-quality input for further analysis. The Phase Lag Index (PLI) was employed to evaluate functional connectivity, a technique that minimizes the impact of volume conduction while measuring phase synchronization. This method offered valuable insights into the brain connectivity patterns that are associated with fatigue and rest.

In order to handle the high dimensionality of the EEG data, feature selection was performed using eleven advanced algorithms, such as LASSO regression, ReliefF, and Recursive Feature Elimination with Correlation Bias Reduction (RFE-CBR). These methods improved the efficacy and interpretability of the model by identifying the most pertinent features. During the classification phase, five machine learning algorithms were implemented: k-Nearest Neighbors (kNN), Support Vector Machines (SVM) with linear and RBF kernels, Linear Discriminant Analysis (LDA), and Random Forest. The purpose of these classifiers was to differentiate between states of fatigue and rest. In order to assess the models' generalizability and performance, the study implemented Leave-One-Out (LOO) cross-validation, which assessed the classifiers across participants. Accuracy, specificity, sensitivity, and F1-score were computed to evaluate the performance of the classification models. LASSO was the most effective feature selection algorithm for this dataset, obtaining a remarkable accuracy of 97.5% with only 19 features. Not only does this minimal yet highly informative feature set guarantee exceptional classification performance, but it also improves the interpretability of the model. The robustness of LASSO is demonstrated by its consistent performance across all classifiers, which achieves ideal performance metrics.

The most relevant features of the PLI Matrix identified by the LASSO feature selection were translated into specific EEG channel connections and their associated frequency bands. The algorithm identified top-ranked features derived from LASSO, mapping them to connections of the PLI matrix. Each feature corresponds to a pair of EEG channels and is linked to one of five frequency bands: Delta, Theta, Alpha, Beta and Gamma. Using a lookup table, the algorithm converts feature indices into channel names and frequency bands in order to provide detailed information on source-target channel pairs and their frequency bands. This reveals functional connectivity patterns, offering insights into how different brain regions interact across frequency bands during rest and fatigue states.

The analysis of frequency band connectivity highlights the dominance of the Theta (4–7 Hz) and Delta (1–4 Hz) bands in distinguishing rest and fatigue states. Theta connectivity, which accounts for 36.8% of the total connections, is strongly associated with cognitive effort and sustained attention, reflecting the brain's compensatory mechanisms to maintain performance during fatigue. The significance of theta oscillations as an fatigue marker is emphasized by their prominence, which is frequently associated with the frontal and central regions where cognitive processing and attentional control are most active [88], [89], [90]. Delta connectivity, contributing 31.6% of the connections, underscores the brain's restorative processes and large-

scale synchronization, indicative of its efforts to recover and manage prolonged cognitive strain. The data's high representation of delta oscillations implies their critical involvement in both physical and mental fatigue, as these oscillations frequently manifest in states that necessitate recovery or rest [91], [92], [93], [94], [95].

The neural activity that distinguishes rest and fatigue states is primarily dominated by the Frontal and Central Lobes, as revealed by the analysis of brain regions implicated in PLI connectivity. The Parietal lobe contributes moderately. The Frontal Lobe, accounting for 42.1% of the connections, emerges as the most significant region in this analysis. This discovery is consistent with its well-established function in executive functions, decision-making, and attentional control, which are processes that are particularly susceptible to the effects of fatigue [36], [37]. The increased connectivity observed in the frontal lobe during fatigue likely represents compensatory mechanisms employed by the brain to sustain task performance despite strained cognitive resources. This elevated activity suggests an effort to mitigate the cognitive slowdown commonly associated with fatigue [33], [43], [52]. The frontal lobe's critical function in facilitating adaptive cognitive responses and sustained attention is underscored by its prominence in this analysis [94], [95], [99], [100].

The Central Lobe, which accounts for 31.6% of the connections, is essential for the integration of sensory and motor information. Its substantial representation emphasizes the significance of sensorimotor processing in the differentiation of fatigue from rest. The central lobe is the primary mediator of these functions, which are frequently characterized by altered sensory responsiveness, reduced motor precision, and delayed reaction times in response to fatigue [101], [102]. The brain's adaptive response to decreased energy levels is reflected in the notable connectivity in this region, which suggests that fatigue-induced changes in motor readiness and sensory processing serve as important markers of neural state alterations [103], [104].

The Parietal Lobe, which accounts for 18.4% of the connections, plays a critical role in the modulation of attention, sensory integration, and spatial awareness. These functions can be substantially impacted by fatigue, which can lead to impaired sensory information processing and reduced spatial accuracy. The parietal lobe's moderate contribution to this analysis underscores its significance in offering valuable insights into the neural adaptations that are induced by fatigue. Its involvement is likely indicative of deviations in the processing and integration of sensory and spatial information, which are critical for maintaining task performance in challenging circumstances [88], [90], [105].

These results collectively illustrate the complex neural dynamics that underlie fatigue and rest, providing a thorough comprehension of the manner in which brain connectivity alters in response to cognitive strain. By focusing on functional connectivity, frequency bands, and regional brain activity, this thesis highlights the critical roles of the Frontal and Central Lobes, as well as the contributions of the Parietal Lobe, in managing and adapting to fatigue. Theta and Delta oscillations' predominant presence further underscores their potential as biomarkers for fatigue detection, as they reflect both compensatory mechanisms and restorative processes.

The potential of EEG-based approaches for reliable fatigue classification is demonstrated by the integration of sophisticated feature selection methods such as LASSO and machine learning classifiers applied in Phase Synchronization metrics (PLI). These insights establish the groundwork for the creation of real-time, non-invasive monitoring systems. These systems have the potential to substantially improve safety and performance in high-stakes environments by identifying fatigue-related risks and facilitating timely interventions. This research emphasizes the significance of interdisciplinary methodologies that integrate neuroscience, computational tools, and machine learning to address complicated challenges in mental state classification.

5 Conclusion and Future Considerations

The primary objective of this research was to develop a machine learning framework that is capable of accurately identifying mental fatigue through the use of EEG data. The study obtained a substantial level of accuracy and interpretability by integrating sophisticated preprocessing techniques, feature selection methods, and robust classifiers. The LASSO feature selection method was the most effective, obtaining a classification accuracy of 97.5% with a minimal feature set. The central and frontal brain regions were emphasized in the findings, as delta and theta frequency bands offered critical insights into the neural dynamics of fatigue.

The findings of this research have significant implications for the scientific comprehension of mental fatigue and its practical applications. The study improves our comprehension of the neural mechanisms that underlie cognitive decline during fatigue by identifying neural markers that are associated with fatigue. The significance of the frontal and central lobes in cognitive control and sensorimotor integration is highlighted by the emphasis on functional connectivity. These results are consistent with the current body of literature, thereby underscoring the importance of delta and theta rhythms in sustained attention and restorative processes.

The methodology proposed in the present research offers a scalable and interpretable framework for fatigue detection from a practical perspective. This has direct applications in a variety of sectors, such as transportation, healthcare, and military operations, where the monitoring of cognitive states is essential for safety and performance. The system is more feasible for deployment in operational environments due to the high accuracy obtained with a minimal feature set, which also enables real-time implementation in widespread EEG devices.

The approach and results of this research are in accordance with the existing literature on EEG-based fatigue detection. The utility of machine learning classifiers in identifying mental fatigue has been demonstrated in numerous prior studies. However, this work distinguishes itself by emphasizing feature selection to improve interpretability without compromising accuracy. The utilization of LASSO regression to attain both high classification performance and dimensionality reduction is a noteworthy accomplishment. Additionally, the emphasis on functional connectivity, particularly through the Phase Lag Index (PLI), offers a more sophisticated comprehension of neural dynamics than research that exclusively relies on spectral power analysis.

This study has certain limitations that necessitate further investigation, despite the promising results. Although the sample size is sufficient for this preliminary investigation, it could be increased to enhance generalizability. The robustness of the findings would be improved by a more diverse and extensive dataset that includes participants from a variety of backgrounds and age categories. The fatigue states examined in this investigation were not quantifiable across multiple levels or gradients. This binary classification of rest versus fatigue may oversimplify the spectrum of fatigue experiences, thereby restricting the framework's capacity to identify minor changes in cognitive states. Furthermore, the controlled experimental design may not completely capture the variability that is present in real-world scenarios. The proposed framework's robustness in operational contexts may be challenged by factors such as environmental noise, variable levels of fatigue, and differences in task complexity, which could influence the EEG signals.

Future research should concentrate on expanding the number of datasets to include a wider range of participant groups and demographics, thereby facilitating validation across a variety of fatigue-inducing conditions and demographics, in an effort to build upon the limitations and findings of this study. It is important to evaluate the robustness and practicality of the proposed framework in dynamic environments by testing it in real-world applications, such as wearable EEG devices for continuous monitoring. Additionally, the accuracy and reliability of fatigue detection systems could be enhanced by the integration of multi-modal data, which involves the combination of

EEG with other physiological signals such as skin conductance or heart rate variability. Further research on the long-term effects of fatigue on cognitive performance and neural connectivity would facilitate the development of predictive models for early fatigue detection, thereby expanding the system's applications in both clinical and operational environments. To further enhance practicality, the use of dry EEG systems could be explored, as they are more suited for real-world applications and reduce setup complexity. Additionally, employing fewer EEG channels concentrated in regions of interest, such as the frontal and central areas, could simplify the system while maintaining accuracy, making it more feasible for wearable and portable solutions.

This thesis introduces a framework for EEG-based fatigue detection that is both interpretable and robust, illustrating the potential of integrating machine learning classifiers with sophisticated feature selection techniques. The results contribute to the expanding corpus of knowledge on mental fatigue and its neural correlates, providing practical solutions for its detection and management. The provided methodology has the potential to facilitate the development of innovative applications in cognitive health monitoring and operational safety across a variety of industries by addressing the enumerated limitations and pursuing the proposed future directions.

Bibliographical References

- [1] J. K. Mai and G. Paxinos, *The human nervous system*. Academic press, 2011.
- [2] E. Niedermeyer and F. H. L. da Silva, *Electroencephalography: Basic Principles, Clinical Applications, and Related Fields*. Lippincott Williams & Wilkins, 2005.
- [3] X. Xing, W. Pei, Y. Wang, Z. Liu, and H. Chen, “Design of High-Density Electrodes For EEG Acquisition,” in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Honolulu, HI: IEEE, Jul. 2018, pp. 1295–1298. doi: 10.1109/EMBC.2018.8512577.
- [4] G. A. Slipher, W. D. Hairston, J. C. Bradford, E. D. Bain, and R. A. Mrozek, “Carbon nanofiber-filled conductive silicone elastomers as soft, dry bioelectronic interfaces,” *PLOS ONE*, vol. 13, no. 2, p. e0189415, Feb. 2018, doi: 10.1371/journal.pone.0189415.
- [5] P. Giacometti, K. L. Perdue, and S. G. Diamond, “Algorithm to find high density EEG scalp coordinates and analysis of their correspondence to structural and functional regions of the brain,” *J. Neurosci. Methods*, vol. 229, pp. 84–96, May 2014, doi: 10.1016/j.jneumeth.2014.04.020.
- [6] E. Başar and A. Düzgün, “The CLAIR model: Extension of Brodmann areas based on brain oscillations and connectivity,” *Int. J. Psychophysiol.*, vol. 103, pp. 185–198, May 2016, doi: 10.1016/j.ijpsycho.2015.02.018.
- [7] C. S. Nayak and A. C. Anilkumar, “EEG Normal Waveforms,” in *StatPearls*, Treasure Island (FL): StatPearls Publishing, 2024. Accessed: Dec. 10, 2024. [Online]. Available: <http://www.ncbi.nlm.nih.gov/books/NBK539805/>
- [8] J. J. Newson and T. C. Thiagarajan, “EEG Frequency Bands in Psychiatric Disorders: A Review of Resting State Studies,” *Front. Hum. Neurosci.*, vol. 12, p. 521, Jan. 2019, doi: 10.3389/fnhum.2018.00521.
- [9] O. Sporns, *Networks of the Brain*. The MIT Press, 2010. doi: 10.7551/mitpress/8476.001.0001.
- [10] D. S. Bassett and O. Sporns, “Network neuroscience,” *Nat. Neurosci.*, vol. 20, no. 3, pp. 353–364, Feb. 2017, doi: 10.1038/nn.4502.
- [11] M. Rubinov and O. Sporns, “Complex network measures of brain connectivity: uses and interpretations,” *NeuroImage*, vol. 52, no. 3, pp. 1059–1069, Sep. 2010, doi: 10.1016/j.neuroimage.2009.10.003.
- [12] M. P. van den Heuvel and O. Sporns, “Rich-club organization of the human connectome,” *J. Neurosci. Off. J. Soc. Neurosci.*, vol. 31, no. 44, pp. 15775–15786, Nov. 2011, doi: 10.1523/JNEUROSCI.3539-11.2011.
- [13] E. Bullmore and O. Sporns, “Complex brain networks: graph theoretical analysis of structural and functional systems,” *Nat. Rev. Neurosci.*, vol. 10, no. 3, pp. 186–198, Mar. 2009, doi: 10.1038/nrn2575.
- [14] S. Achard and E. Bullmore, “Efficiency and cost of economical brain functional networks,” *PLoS Comput. Biol.*, vol. 3, no. 2, p. e17, Feb. 2007, doi: 10.1371/journal.pcbi.0030017.
- [15] D. J. Watts and S. H. Strogatz, “Collective dynamics of ‘small-world’ networks,” *Nature*, vol. 393, no. 6684, pp. 440–442, Jun. 1998, doi: 10.1038/30918.
- [16] P. Hagmann *et al.*, “Mapping the structural core of human cerebral cortex,” *PLoS Biol.*, vol. 6, no. 7, p. e159, Jul. 2008, doi: 10.1371/journal.pbio.0060159.
- [17] M. E. J. Newman, “The Structure and Function of Complex Networks,” *SIAM Rev.*, vol. 45, no. 2, pp. 167–256, Jan. 2003, doi: 10.1137/S003614450342480.
- [18] D. S. Bassett and E. Bullmore, “Small-World Brain Networks,” *The Neuroscientist*, vol. 12, no. 6, pp. 512–523, Dec. 2006, doi: 10.1177/1073858406293182.
- [19] M. Kaiser and C. C. Hilgetag, “Nonoptimal Component Placement, but Short Processing Paths, due to Long-Distance Projections in Neural Systems,” *PLoS Comput. Biol.*, vol. 2, no. 7, p. e95, Jul. 2006, doi: 10.1371/journal.pcbi.0020095.
- [20] M. P. Van Den Heuvel and O. Sporns, “Network hubs in the human brain,” *Trends Cogn. Sci.*, vol. 17, no. 12, pp. 683–696, Dec. 2013, doi: 10.1016/j.tics.2013.09.012.

- [21] R. L. Buckner *et al.*, “Cortical Hubs Revealed by Intrinsic Functional Connectivity: Mapping, Assessment of Stability, and Relation to Alzheimer’s Disease,” *J. Neurosci.*, vol. 29, no. 6, pp. 1860–1873, Feb. 2009, doi: 10.1523/JNEUROSCI.5062-08.2009.
- [22] D. S. Bassett and M. S. Gazzaniga, “Understanding complexity in the human brain,” *Trends Cogn. Sci.*, vol. 15, no. 5, pp. 200–209, May 2011, doi: 10.1016/j.tics.2011.03.006.
- [23] Y. He, Z. J. Chen, and A. C. Evans, “Small-world anatomical networks in the human brain revealed by cortical thickness from MRI,” *Cereb. Cortex N. Y. N 1991*, vol. 17, no. 10, pp. 2407–2419, Oct. 2007, doi: 10.1093/cercor/bhl149.
- [24] F. X. Castellanos and Y. Aoki, “Intrinsic Functional Connectivity in Attention-Deficit/Hyperactivity Disorder: A Science in Development,” *Biol. Psychiatry Cogn. Neurosci. Neuroimaging*, vol. 1, no. 3, pp. 253–261, May 2016, doi: 10.1016/j.bpsc.2016.03.004.
- [25] S. Whitfield-Gabrieli and J. M. Ford, “Default Mode Network Activity and Connectivity in Psychopathology,” *Annu. Rev. Clin. Psychol.*, vol. 8, no. 1, pp. 49–76, Apr. 2012, doi: 10.1146/annurev-clinpsy-032511-143049.
- [26] L. Kester and P. A. Kirschner, “Cognitive Tasks and Learning,” in *Encyclopedia of the Sciences of Learning*, N. M. Seel, Ed., Boston, MA: Springer US, 2012, pp. 619–622. doi: 10.1007/978-1-4419-1428-6_225.
- [27] A. Skulmowski and G. D. Rey, “Measuring Cognitive Load in Embodied Learning Settings,” *Front. Psychol.*, vol. 8, p. 1191, Aug. 2017, doi: 10.3389/fpsyg.2017.01191.
- [28] M. S. Young, K. A. Brookhuis, C. D. Wickens, and P. A. Hancock, “State of science: mental workload in ergonomics,” *Ergonomics*, vol. 58, no. 1, pp. 1–17, Jan. 2015, doi: 10.1080/00140139.2014.956151.
- [29] A. Seidler *et al.*, “The role of psychosocial working conditions on burnout and its core component emotional exhaustion – a systematic review,” *J. Occup. Med. Toxicol.*, vol. 9, no. 1, p. 10, 2014, doi: 10.1186/1745-6673-9-10.
- [30] Y. Lean and F. Shan, “Brief review on physiological and biochemical evaluations of human mental workload,” *Hum. Factors Ergon. Manuf. Serv. Ind.*, vol. 22, no. 3, pp. 177–187, May 2012, doi: 10.1002/hfm.20269.
- [31] G. Borghini, L. Astolfi, G. Vecchiato, D. Mattia, and F. Babiloni, “Measuring neurophysiological signals in aircraft pilots and car drivers for the assessment of mental workload, fatigue and drowsiness,” *Neurosci. Biobehav. Rev.*, vol. 44, pp. 58–75, Jul. 2014, doi: 10.1016/j.neubiorev.2012.10.003.
- [32] E. Niedermeyer and F. H. Lopes da Silva, Eds., *Electroencephalography: basic principles, clinical applications, and related fields*, 5th ed. Philadelphia: Lippincott Williams & Wilkins, 2005.
- [33] P. Antonenko, F. Paas, R. Grabner, and T. Van Gog, “Using Electroencephalography to Measure Cognitive Load,” *Educ. Psychol. Rev.*, vol. 22, no. 4, pp. 425–438, Dec. 2010, doi: 10.1007/s10648-010-9130-y.
- [34] C. MÅ¼hl, C. Jeunet, and F. Lotte, “EEG-based workload estimation across affective contexts,” *Front. Neurosci.*, vol. 8, Jun. 2014, doi: 10.3389/fnins.2014.00114.
- [35] Y. Chen and X. Huang, “Modulation of Alpha and Beta Oscillations during an n-back Task with Varying Temporal Memory Load,” *Front. Psychol.*, vol. 6, Jan. 2016, doi: 10.3389/fpsyg.2015.02031.
- [36] T. Radüntz, “The Effect of Planning, Strategy Learning, and Working Memory Capacity on Mental Workload,” *Sci. Rep.*, vol. 10, no. 1, p. 7096, Apr. 2020, doi: 10.1038/s41598-020-63897-6.
- [37] S. Puma, N. Matton, P.-V. Paubel, É. Raufaste, and R. El-Yagoubi, “Using theta and alpha band power to assess cognitive workload in multitasking environments,” *Int. J. Psychophysiol.*, vol. 123, pp. 111–120, Jan. 2018, doi: 10.1016/j.ijpsycho.2017.10.004.
- [38] M. Spüler, C. Walter, W. Rosenstiel, P. Gerjets, K. Moeller, and E. Klein, “EEG-based prediction of cognitive workload induced by arithmetic: a step towards online adaptation in numerical learning,” *ZDM*, vol. 48, no. 3, pp. 267–278, Jun. 2016, doi: 10.1007/s11858-015-0754-8.

- [39] P. Zarjam, J. Epps, F. Chen, and N. H. Lovell, "Estimating cognitive workload using wavelet entropy-based features during an arithmetic task," *Comput. Biol. Med.*, vol. 43, no. 12, pp. 2186–2195, Dec. 2013, doi: 10.1016/j.combiomed.2013.08.021.
- [40] R. N. Roy, S. Charbonnier, A. Campagne, and S. Bonnet, "Efficient mental workload estimation using task-independent EEG features," *J. Neural Eng.*, vol. 13, no. 2, p. 026019, Apr. 2016, doi: 10.1088/1741-2560/13/2/026019.
- [41] S. Wang, J. Gwizdka, and W. A. Chaovalitwongse, "Using Wireless EEG Signals to Assess Memory Workload in the n -Back Task," *IEEE Trans. Hum.-Mach. Syst.*, vol. 46, no. 3, pp. 424–435, Jun. 2016, doi: 10.1109/THMS.2015.2476818.
- [42] C. Walter, S. Schmidt, W. Rosenstiel, P. Gerjets, and M. Bogdan, "Using Cross-Task Classification for Classifying Workload Levels in Complex Learning Tasks," in *2013 Humaine Association Conference on Affective Computing and Intelligent Interaction*, Geneva, Switzerland: IEEE, Sep. 2013, pp. 876–881. doi: 10.1109/ACII.2013.164.
- [43] C. L. Baldwin and B. N. Penaranda, "Adaptive training using an artificial neural network and EEG metrics for within- and cross-task workload classification," *NeuroImage*, vol. 59, no. 1, pp. 48–56, Jan. 2012, doi: 10.1016/j.neuroimage.2011.07.047.
- [44] Y. Ke *et al.*, "Towards an effective cross-task mental workload recognition model using electroencephalography based on feature selection and support vector machine regression," *Int. J. Psychophysiol.*, vol. 98, no. 2, pp. 157–166, Nov. 2015, doi: 10.1016/j.ijpsycho.2015.10.004.
- [45] A. Gevins *et al.*, "Monitoring Working Memory Load during Computer-Based Tasks with EEG Pattern Recognition Methods," *Hum. Factors J. Hum. Factors Ergon. Soc.*, vol. 40, no. 1, pp. 79–91, Mar. 1998, doi: 10.1518/001872098779480578.
- [46] G. Zhao, Y.-J. Liu, and Y. Shi, "Real-Time Assessment of the Cross-Task Mental Workload Using Physiological Measures During Anomaly Detection," *IEEE Trans. Hum.-Mach. Syst.*, vol. 48, no. 2, pp. 149–160, Apr. 2018, doi: 10.1109/THMS.2018.2803025.
- [47] P. Zhang, X. Wang, W. Zhang, and J. Chen, "Learning Spatial–Spectral–Temporal EEG Features With Recurrent 3D Convolutional Neural Networks for Cross-Task Mental Workload Assessment," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 27, no. 1, pp. 31–42, Jan. 2019, doi: 10.1109/TNSRE.2018.2884641.
- [48] E. Debie *et al.*, "Multimodal Fusion for Objective Assessment of Cognitive Workload: A Review," *IEEE Trans. Cybern.*, vol. 51, no. 3, pp. 1542–1555, Mar. 2021, doi: 10.1109/TCYB.2019.2939399.
- [49] D. S. Bassett and O. Sporns, "Network neuroscience," *Nat. Neurosci.*, vol. 20, no. 3, pp. 353–364, Mar. 2017, doi: 10.1038/nn.4502.
- [50] M. G. Kitzbichler, R. N. A. Henson, M. L. Smith, P. J. Nathan, and E. T. Bullmore, "Cognitive Effort Drives Workspace Configuration of Human Brain Functional Networks," *J. Neurosci.*, vol. 31, no. 22, pp. 8259–8270, Jun. 2011, doi: 10.1523/JNEUROSCI.0440-11.2011.
- [51] M. A. Klados *et al.*, "A Graph Theoretical Approach to Study the Organization of the Cortical Networks during Different Mathematical Tasks," *PLoS ONE*, vol. 8, no. 8, p. e71800, Aug. 2013, doi: 10.1371/journal.pone.0071800.
- [52] G. N. Dimitrakopoulos *et al.*, "Functional Connectivity Analysis of Mental Fatigue Reveals Different Network Topological Alterations Between Driving and Vigilance Tasks," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 26, no. 4, pp. 740–749, Apr. 2018, doi: 10.1109/TNSRE.2018.2791936.
- [53] S. I. Dimitriadis, Y. Sun, K. Kwok, N. A. Laskaris, N. Thakor, and A. Bezerianos, "Cognitive Workload Assessment Based on the Tensorial Treatment of EEG Estimates of Cross-Frequency Phase Interactions," *Ann. Biomed. Eng.*, vol. 43, no. 4, pp. 977–989, Apr. 2015, doi: 10.1007/s10439-014-1143-0.
- [54] I. Kakkos *et al.*, "Mental Workload Drives Different Reorganizations of Functional Cortical Connectivity Between 2D and 3D Simulated Flight Experiments," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 27, no. 9, pp. 1704–1713, Sep. 2019, doi: 10.1109/TNSRE.2019.2930082.
- [55] G. N. Dimitrakopoulos *et al.*, "Task-Independent Mental Workload Classification Based Upon Common Multiband EEG Cortical Connectivity," *IEEE Trans. Neural Syst.*

- Rehabil. Eng.*, vol. 25, no. 11, pp. 1940–1949, Nov. 2017, doi: 10.1109/TNSRE.2017.2701002.
- [56] A. Delorme and S. Makeig, “EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis,” *J. Neurosci. Methods*, vol. 134, no. 1, pp. 9–21, Mar. 2004, doi: 10.1016/j.jneumeth.2003.10.009.
- [57] M. D. Lutovac, D. V. Tošić, and B. L. Evans, *Filter design for signal processing using MATLAB and Mathematica*. Upper Saddle River, N.J.: Prentice Hall, 2001.
- [58] G. Roffo, S. Melzi, U. Castellani, and A. Vinciarelli, “Infinite Latent Feature Selection: A Probabilistic Latent Graph-Based Ranking Approach,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice: IEEE, Oct. 2017, pp. 1407–1415. doi: 10.1109/ICCV.2017.156.
- [59] M. Robnik-Šikonja and I. Kononenko, “Theoretical and Empirical Analysis of ReliefF and RReliefF,” *Mach. Learn.*, vol. 53, no. 1/2, pp. 23–69, 2003, doi: 10.1023/A:1025667309714.
- [60] I. Kononenko, “Estimating attributes: Analysis and extensions of RELIEF,” in *Machine Learning: ECML-94*, vol. 784, F. Bergadano and L. Raedt, Eds., in Lecture Notes in Computer Science, vol. 784. , Berlin, Heidelberg: Springer Berlin Heidelberg, 1994, pp. 171–182. doi: 10.1007/3-540-57868-4_57.
- [61] M. Beraha, A. M. Metelli, M. Papini, A. Tirinzoni, and M. Restelli, “Feature Selection via Mutual Information: New Theoretical Insights,” 2019, *arXiv*. doi: 10.48550/ARXIV.1907.07384.
- [62] J. R. Vergara and P. A. Estévez, “A review of feature selection methods based on mutual information,” *Neural Comput. Appl.*, vol. 24, no. 1, pp. 175–186, Jan. 2014, doi: 10.1007/s00521-013-1368-0.
- [63] Y.-W. Chen and C.-J. Lin, “Combining SVMs with Various Feature Selection Strategies,” in *Feature Extraction*, vol. 207, 2008, pp. 315–324. doi: 10.1007/978-3-540-35488-8_13.
- [64] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik, “Feature Selection for SVMs,” in *Advances in Neural Information Processing Systems*, T. Leen, T. Dietterich, and V. Tresp, Eds., MIT Press, 2000. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2000/file/8c3039bd5842dca3d944faab91447818-Paper.pdf
- [65] X. He, D. Cai, and P. Niyogi, “Laplacian Score for Feature Selection,” in *Advances in Neural Information Processing Systems*, Y. Weiss, B. Schölkopf, and J. Platt, Eds., MIT Press, 2005. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2005/file/b5b03f06271f8917685d14cea7c6c50a-Paper.pdf
- [66] L. Zhu, L. Miao, and D. Zhang, “Iterative Laplacian Score for Feature Selection,” in *Pattern Recognition*, vol. 321, C.-L. Liu, C. Zhang, and L. Wang, Eds., in Communications in Computer and Information Science, vol. 321. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 80–87. doi: 10.1007/978-3-642-33506-8_11.
- [67] M. Gan and L. Zhang, “Iteratively local fisher score for feature selection,” *Appl. Intell.*, vol. 51, no. 8, pp. 6167–6181, Aug. 2021, doi: 10.1007/s10489-020-02141-0.
- [68] Q. Gu, Z. Li, and J. Han, “Generalized Fisher Score for Feature Selection,” 2012, *arXiv*. doi: 10.48550/ARXIV.1202.3725.
- [69] H. Zeng and Y. Cheung, “Feature Selection for Local Learning Based Clustering,” in *Advances in Knowledge Discovery and Data Mining*, vol. 5476, T. Theeramunkong, B. Kijssirikul, N. Cercone, and T.-B. Ho, Eds., in Lecture Notes in Computer Science, vol. 5476. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 414–425. doi: 10.1007/978-3-642-01307-2_38.
- [70] Hong Zeng and Yiu-ming Cheung, “Feature Selection and Kernel Learning for Local Learning-Based Clustering,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1532–1547, Aug. 2011, doi: 10.1109/TPAMI.2010.215.
- [71] M. Hall, “Correlation-based Feature Selection for Machine Learning,” 2003. Accessed: Dec. 11, 2024. [Online]. Available: <https://www.semanticscholar.org/paper/Correlation-based-Feature-Selection-for-Machine-Hall/6bc43977fb11ceed0b9aa55b23c6dd29dd9a132>

- [72] T. Hastie, R. Tibshirani, and J. Friedman, "Linear Methods for Regression," in *The Elements of Statistical Learning*, in Springer Series in Statistics. , New York, NY: Springer New York, 2009, pp. 43–99. doi: 10.1007/978-0-387-84858-7_3.
- [73] R. Tibshirani, "Regression Shrinkage and Selection via the Lasso," *J. R. Stat. Soc. Ser. B Methodol.*, vol. 58, no. 1, pp. 267–288, 1996.
- [74] D. Zhang, D. Guo, and K. Yan, "Feature Selection and Analysis on Correlated Breath Data," in *Breath Analysis for Medical Applications*, Singapore: Springer Singapore, 2017, pp. 181–206. doi: 10.1007/978-981-10-4322-2_10.
- [75] K. Yan and D. Zhang, "Feature selection and analysis on correlated gas sensor data with recursive feature elimination," *Sens. Actuators B Chem.*, vol. 212, pp. 353–363, Jun. 2015, doi: 10.1016/j.snb.2015.02.025.
- [76] S. J. Russell and P. Norvig, *Artificial intelligence: a modern approach*, Third edition, Global edition. in Prentice Hall series in artificial intelligence. Boston Columbus Indianapolis: Pearson, 2016.
- [77] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inf. Theory*, vol. 13, no. 1, pp. 21–27, Jan. 1967, doi: 10.1109/TIT.1967.1053964.
- [78] B. Matérn, *Spatial Variation*, vol. 36. in Lecture Notes in Statistics, vol. 36. New York, NY: Springer New York, 1986. doi: 10.1007/978-1-4615-7892-5.
- [79] C. J. C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Min. Knowl. Discov.*, vol. 2, no. 2, pp. 121–167, 1998, doi: 10.1023/A:1009715923555.
- [80] E. Osuna, R. Freund, and F. Girosit, "Training support vector machines: an application to face detection," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Juan, Puerto Rico: IEEE Comput. Soc, 1997, pp. 130–136. doi: 10.1109/CVPR.1997.609310.
- [81] G. Fernandez, "Discriminant Analysis, A Powerful Classification Technique in Data Mining," *Proc SAS Int Conf*, Jan. 2002.
- [82] R. A. Fisher, "THE USE OF MULTIPLE MEASUREMENTS IN TAXONOMIC PROBLEMS," *Ann. Eugen.*, vol. 7, no. 2, pp. 179–188, Sep. 1936, doi: 10.1111/j.1469-1809.1936.tb02137.x.
- [83] L. Breiman, "Random Forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: 10.1023/A:1010933404324.
- [84] A. Liaw and M. Wiener, "Classification and Regression by RandomForest," *Forest*, vol. 23, Nov. 2001.
- [85] C. Cortes and V. Vapnik, "Support-Vector Networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995, doi: 10.1023/A:1022627411411.
- [86] Y. Freund and R. E. Schapire, "Large Margin Classification Using the Perceptron Algorithm," *Mach. Learn.*, vol. 37, no. 3, pp. 277–296, Dec. 1999, doi: 10.1023/A:1007662407062.
- [87] D. L. Olson and D. Delen, "Performance Evaluation for Predictive Modeling," in *Advanced Data Mining Techniques*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 137–147. doi: 10.1007/978-3-540-76917-0_9.
- [88] A. Ardestani, W. Shen, F. Darvas, A. W. Toga, and J. M. Fuster, "Modulation of Frontoparietal Neurovascular Dynamics in Working Memory," *J. Cogn. Neurosci.*, vol. 28, no. 3, pp. 379–401, Mar. 2016, doi: 10.1162/jocn_a_00903.
- [89] C. Babiloni *et al.*, "Human cortical responses during one-bit short-term memory. A high-resolution EEG study on delayed choice reaction time tasks," *Clin. Neurophysiol.*, vol. 115, no. 1, pp. 161–170, Jan. 2004, doi: 10.1016/S1388-2457(03)00286-4.
- [90] P. S. Cooper *et al.*, "Theta frontoparietal connectivity associated with proactive and reactive cognitive control processes," *NeuroImage*, vol. 108, pp. 354–363, Mar. 2015, doi: 10.1016/j.neuroimage.2014.12.028.
- [91] R. Gulbinaite, H. Van Rijn, and M. X. Cohen, "Fronto-parietal network oscillations reveal relationship between working memory capacity and cognitive control," *Front. Hum. Neurosci.*, vol. 8, Sep. 2014, doi: 10.3389/fnhum.2014.00761.
- [92] Y. Liu, H. Ayaz, and P. A. Shewokis, "Multisubject 'Learning' for Mental Workload Classification Using Concurrent EEG, fNIRS, and Physiological Measures," *Front. Hum. Neurosci.*, vol. 11, p. 389, Jul. 2017, doi: 10.3389/fnhum.2017.00389.

- [93] S. Hanouneh, H. U. Amin, N. M. Saad, and A. S. Malik, "EEG Power and Functional Connectivity Correlates with Semantic Long-Term Memory Retrieval," *IEEE Access*, vol. 6, pp. 8695–8703, 2018, doi: 10.1109/ACCESS.2017.2788859.
- [94] P. Zarjam, J. Epps, and N. H. Lovell, "Beyond Subjective Self-Rating: EEG Signal Classification of Cognitive Workload," *IEEE Trans. Auton. Ment. Dev.*, vol. 7, no. 4, pp. 301–310, Dec. 2015, doi: 10.1109/TAMD.2015.2441960.
- [95] H. Kondo, M. Morishita, N. Osaka, M. Osaka, H. Fukuyama, and H. Shibasaki, "Functional roles of the cingulo-frontal network in performance on working memory," *NeuroImage*, vol. 21, no. 1, pp. 2–14, Jan. 2004, doi: 10.1016/j.neuroimage.2003.09.046.
- [96] D. Huang *et al.*, "Combining Partial Directed Coherence and Graph Theory to Analyse Effective Brain Networks of Different Mental Tasks," *Front. Hum. Neurosci.*, vol. 10, May 2016, doi: 10.3389/fnhum.2016.00235.
- [97] M. G. Kitzbichler, R. N. A. Henson, M. L. Smith, P. J. Nathan, and E. T. Bullmore, "Cognitive Effort Drives Workspace Configuration of Human Brain Functional Networks," *J. Neurosci.*, vol. 31, no. 22, pp. 8259–8270, Jun. 2011, doi: 10.1523/JNEUROSCI.0440-11.2011.
- [98] R. Vijayalakshmi, D. Nandagopal, N. Dasari, B. Cocks, N. Dahal, and M. Thilaga, "Minimum connected component – A novel approach to detection of cognitive load induced changes in functional brain networks," *Neurocomputing*, vol. 170, pp. 15–31, Dec. 2015, doi: 10.1016/j.neucom.2015.03.092.
- [99] A.-M. Brouwer, M. A. Hogervorst, J. B. F. Van Erp, T. Heffelaar, P. H. Zimmerman, and R. Oostenveld, "Estimating workload using EEG spectral power and ERPs in the n-back task," *J. Neural Eng.*, vol. 9, no. 4, p. 045008, Aug. 2012, doi: 10.1088/1741-2560/9/4/045008.
- [100] I. Seleznev *et al.*, "Detrended Fluctuation, Coherence, and Spectral Power Analysis of Activation Rearrangement in EEG Dynamics During Cognitive Workload," *Front. Hum. Neurosci.*, vol. 13, p. 270, Aug. 2019, doi: 10.3389/fnhum.2019.00270.
- [101] S. Nieuwenhuis, D. J. Heslenfeld, N. J. Alting Von Geusau, R. B. Mars, C. B. Holroyd, and N. Yeung, "Activity in human reward-sensitive brain areas is strongly context dependent," *NeuroImage*, vol. 25, no. 4, pp. 1302–1309, May 2005, doi: 10.1016/j.neuroimage.2004.12.043.
- [102] R. G. O'Connell *et al.*, "The role of cingulate cortex in the detection of errors with and without awareness: a high-density electrical mapping study," *Eur. J. Neurosci.*, vol. 25, no. 8, pp. 2571–2579, Apr. 2007, doi: 10.1111/j.1460-9568.2007.05477.x.
- [103] P. Ahmadian, S. Cagnoni, and L. Ascari, "How capable is non-invasive EEG data of predicting the next movement? A mini review," *Front. Hum. Neurosci.*, vol. 7, 2013, doi: 10.3389/fnhum.2013.00124.
- [104] A. Shakeel, M. S. Navid, M. N. Anwar, S. Mazhar, M. Jochumsen, and I. K. Niazi, "A Review of Techniques for Detection of Movement Intention Using Movement-Related Cortical Potentials," *Comput. Math. Methods Med.*, vol. 2015, pp. 1–13, 2015, doi: 10.1155/2015/346217.

Appendix

Matlab Code

Main Program

```
%% Main folders and paths
addpath 'C:\MASTER TEAM\Thesis\eeqlab2024.2\'
savepath='C:\MASTER TEAM\Thesis\output\';
addpath(genpath('C:\MASTER
TEAM\Thesis\network_PLI_and_metrics\FSLib_v7.0.1_2020_2'))
addpath(genpath('C:\MASTER TEAM\Thesis\network_PLI_and_metrics\libsvm-
3.23'))
addpath(genpath('C:\MASTER TEAM\Thesis\network_PLI_and_metrics\rfe'))
Bandrange=[1 4;4 7;8 12;13 30;31 45];

%% Read and Analyze Rest Data
addpath 'C:\MASTER TEAM\Thesis\Rest\'
f=dir('C:\MASTER TEAM\Thesis\Rest\*.mat');
loadpath='C:\MASTER TEAM\Thesis\Rest\';

AllSubPLI={};
AllsesbandPLI={};

for i=1:length(f)
    eeglab
    name2load = f(i).name;
    load(name2load)
    EEG = EEG_NB;
    eeglab redraw
        for band=1:size(Bandrange,1)
            switch band
                case 1
                    EEGtemp = pop_eegfiltnew(EEG, Bandrange(band,1),
Bandrange(band,2), 2640, 0, [], 0);
                case 2
                    EEGtemp = pop_eegfiltnew(EEG, Bandrange(band,1),
Bandrange(band,2), 660, 0, [], 0);
                case 3
                    EEGtemp = pop_eegfiltnew(EEG, Bandrange(band,1),
Bandrange(band,2), 660, 0, [], 0);
                case 4
                    EEGtemp = pop_eegfiltnew(EEG, Bandrange(band,1),
Bandrange(band,2), 408, 0, [], 0);
                case 5
                    EEGtemp = pop_eegfiltnew(EEG, Bandrange(band,1),
Bandrange(band,2), 172, 0, [], 0);
            end
            data=EEGtemp.data;

            PLI = PLI_Cal(data);
            AllsesbandPLI{band,1}=PLI;
        end
        AllSubPLI{i,1}=AllsesbandPLI;
end
% calculate the mean every subject across all trials
MeanAllSubPLI={};
for i=1:20
    for j=1:5
        A1=AllSubPLI{i}{j};
```

```

MeanAllSubPLI{i,1}{j,1}=mean(A1,3);
    end
end
MeanAllSubPLI_R=MeanAllSubPLI;
% Convert the Mean PLI Array to 1d array
for i=1:20
    for j=1:5
AA=MeanAllSubPLI_R{i}{j};
a1 = reshape(triu(AA, 1)', 1, []);
a3(j,:) = a1(a1 ~= 0);
        end
        Feat_R(i,:)=reshape(a3',1,[]);
    end
clear a1 A1 a3 AA ALLCOM ALLEEG AllsesbandPLI AllSubPLI CorrectAns
CURRENTSET CURRENTSTUDY data EEG EEG_NB EEGtemp f globalvars i j LASTCOM
loadpath MeanAllSubPLI name2load NB_RT PLI PLUGINLIST STUDY tmpEEG
%% Read and Analyze Fatigue Data
addpath 'C:\MASTER TEAM\Thesis\Fatigue\'
f=dir('C:\MASTER TEAM\Thesis\Fatigue\*.mat');
loadpath='C:\MASTER TEAM\Thesis\Fatigue\'

AllSubPLI={};
AllsesbandPLI={};

for i=1:length(f)
    eeglab
    name2load = f(i).name;
    load(name2load)
    EEG = EEG_NB;
    eeglab redraw
        for band=1:size(Bandrange,1)
            switch band
                case 1
                    EEGtemp = pop_eegfiltnew(EEG, Bandrange(band,1),
Bandrange(band,2), 2640, 0, [], 0);
                case 2
                    EEGtemp = pop_eegfiltnew(EEG, Bandrange(band,1),
Bandrange(band,2), 660, 0, [], 0);
                case 3
                    EEGtemp = pop_eegfiltnew(EEG, Bandrange(band,1),
Bandrange(band,2), 660, 0, [], 0);
                case 4
                    EEGtemp = pop_eegfiltnew(EEG, Bandrange(band,1),
Bandrange(band,2), 408, 0, [], 0);
                case 5
                    EEGtemp = pop_eegfiltnew(EEG, Bandrange(band,1),
Bandrange(band,2), 172, 0, [], 0);
            end
            data=EEGtemp.data;
            PLI = PLI_Cal(data);
            AllsesbandPLI{band,1}=PLI;
        end
        AllSubPLI{i,1}=AllsesbandPLI;
    end
end
% calculate the mean every subject across all trials
MeanAllSubPLI={};
for i=1:20
    for j=1:5
A1=AllSubPLI{i}{j};
MeanAllSubPLI{i,1}{j,1}=mean(A1,3);
        end
    end
end

```

```

MeanAllSubPLI_F=MeanAllSubPLI;
% Convert the Mean PLI Array to 1d array
for i=1:20
    for j=1:5
AA=MeanAllSubPLI_F{i}{j};
a1 = reshape(triu(AA, 1)', 1, []);
a3(j,:) = a1(a1 ~= 0);
        end
        Feat_F(i,:)=reshape(a3',1,[]);
    end
clear a1 A1 a3 AA ALLCOM ALLEEG AllsesbandPLI AllSubPLI CorrectAns
CURRENTSET CURRENTSTUDY data EEG EEG_NB EEGtemp f globalvars i j LASTCOM
loadpath MeanAllSubPLI name2load NB_RT PLI PLUGINLIST STUDY tmpEEG
%% Feature Selection

RR= horzcat(zeros(20,1),Feat_R);
FF=horzcat(ones(20,1),Feat_F);
features=vertcat(RR,FF);
labels = features(:,1);
features(:,1) = [];
CV(:,1)=1:20;
CV(:,2)=21:40;
clear RR FF

%% Feature selection
[FS_all_ranks_dev] = FS_kakkosP(features,labels,CV);

%% PLI
function PLI=PLI_Cal(data)
%points*chan*trials
[cn,pn,tn]=size(data);
PLI=zeros(cn,cn,tn);
for j=1:tn
    signal=data(:, :, j)';
    phaseSignal = angle(hilbert(signal));
    pli=zeros(cn);
    for i = 1:(cn-1)
        for m = (i+1):cn
            pli(i,m) = abs(mean(sign(phaseSignal(:,i)-phaseSignal(:,m))));
        end
    end
    pli = triu(pli);
    pli = pli+pli';
    PLI(:, :, j)=pli;
end
end

```

Feature Selection

```
function [FS_all_ranks] = FS_kakkosP(features, labels, cv)
    % Features: instances(subjects) x features
    % Labels: correspond to instances
    % CV: matrix corresponding to the subjects left for testing in each
    fold

    Instances = size(features,1);
    FeatNo = size(features,2);

    % Convert labels
    Y = labels;
    Y2 = num2str(Y);
    Y3 = nominal(ismember(Y2, '1'));
    Y_train = (double(Y3) - 1)*2 - 1; % labels: neg_class -1, pos_class +1

    numMethods = 11; % Total number of methods
    FS_all_ranks = zeros(numMethods, size(cv,1), FeatNo); % Dimensions:
    [methods x folds x features]

    % Loop over methods
    for M = 1:numMethods

        % Start the parallel loop over folds
        parfor JJ = 1:size(cv,1)
            % Prepare training data
            X_train = features;
            X_train(cv(JJ,:), :) = [];
            Y_train = labels;
            Y_train(cv(JJ,:)) = [];

            switch M
                case 1
                    % ILFS
                    [FS_ranks, ~] = ILFS(X_train, Y_train, 6, 1);
                case 2
                    % ReliefF
                    [FS_ranks, ~] = reliefF(X_train, Y_train, 20);
                case 3
                    % mutinfFs
                    numF = size(X_train,2);
                    [FS_ranks, ~] = mutInfFS(X_train, Y_train, numF);
                case 4
                    % fsv
                    numF = size(X_train,2);
                    [FS_ranks, ~] = fsvFS(X_train, Y_train, numF);
                case 5
                    % Laplacian Score
                    W = dist(X_train');
                    W = -W./max(max(W)); % Similarity matrix
                    [lscores] = LaplacianScore(X_train, W);
                    [~, FS_ranks] = sort(-lscores);
                case 6
                    % Fisher Score
                    numF = size(X_train,2);
                    FS_ranks = spider_wrapper(X_train, Y_train, numF,
'fisher');
                case 7
                    % LLCFS
                    FS_ranks = llcfs(X_train);
```

```

    case 8
        % CFS
        FS_ranks = cfs(X_train);
    case 9
        % LASSO
        lambda = 25;
        B = lasso(X_train, Y_train);
        [~, FS_ranks] = sort(B(:,lambda), 'descend');
    case 10
        % RFC-CBR-linear
        [FS_ranks, ~] = ftSel_SVMRFECBR_ori(X_train, Y_train,
{1; 'true';0.9});
    case 11
        % RFC-CBR-gaussian
        [FS_ranks, ~] = ftSel_SVMRFECBR(X_train, Y_train,
{1; 'true';0.9});
    end
    FS_all_ranks(M, JJ, :) = FS_ranks;
end
end
end

```

Classification

```
%% Classification

cv=CV;
FS_all_ranks = FS_all_ranks_dev;
for ii = 1:size(FS_all_ranks,1)

    for FeatNum = 1:500%size(FS_all_ranks,3)
        tempR = squeeze(FS_all_ranks(ii, :, :));
        Feats = [];
        temp2 = [];
        temp3 = [];
        temp2 = tempR(:, 1:FeatNum);
        for fold = 1:size(FS_all_ranks,2)
            temp3 = temp2(fold, :);
            Feats = horzcat(Feats, temp3);
        end
        [n, bin] = hist(Feats, unique(Feats));
        [~, idx] = sort(-n);
        b = n(idx); % count instances
        a = bin(idx); % corresponding values
        sortedFEATS = a(1:FeatNum);
        cm2 = zeros(2);
        for LOO = 1:size(cv,1)
            NewDataTest = features(cv(LOO, :), sortedFEATS(1:FeatNum));
            TestLabels = labels(cv(LOO, :), :);
            NewDataTrain = features(:, sortedFEATS(1:FeatNum));
            NewDataTrain(cv(LOO, :), :) = [];
            TrainLabels = labels;
            TrainLabels(cv(LOO, :), :) = [];

            KNN = fitcknn(NewDataTrain, TrainLabels, 'NumNeighbors', 3);
            pred = predict(KNN, NewDataTest);
            acc(ii, 1, LOO, FeatNum) =
sum(pred==TestLabels)/size(TestLabels,1);
            TP = size(find(TestLabels == 0 & pred==0),1);
            TN = size(find(TestLabels == 1 & pred==1),1);
            FP = size(find(TestLabels == 1 & pred==0),1);
            FN = size(find(TestLabels == 0 & pred==1),1);
            sensitivity(ii, 1, LOO, FeatNum) = TP / (TP + FN);
            specificity(ii, 1, LOO, FeatNum) = TN / (TN + FP);
            precision(ii, 1, LOO, FeatNum) = TP / (TP + FP);
            f1_score(ii, 1, LOO, FeatNum) = 2 *TP/(2*TP+FP+FN);

            c1 = fitsvm(NewDataTrain, TrainLabels, 'KernelFunction', 'rbf');
            predSVM = predict(c1, NewDataTest);
            acc(ii, 2, LOO, FeatNum) =
sum(predSVM==TestLabels)/size(TestLabels,1);
            TP = size(find(TestLabels == 0 & predSVM==0),1);
            TN = size(find(TestLabels == 1 & predSVM==1),1);
            FP = size(find(TestLabels == 1 & predSVM==0),1);
            FN = size(find(TestLabels == 0 & predSVM==1),1);
            sensitivity(ii, 2, LOO, FeatNum) = TP / (TP + FN);
            specificity(ii, 2, LOO, FeatNum) = TN / (TN + FP);
            precision(ii, 2, LOO, FeatNum) = TP / (TP + FP);
            f1_score(ii, 2, LOO, FeatNum) = 2 *TP/(2*TP+FP+FN);

            Mdllinear = fitcdiscr(NewDataTrain, TrainLabels);
            predLDA = predict(Mdllinear, NewDataTest);
```

```

        acc(ii,3,LOO,FeatNum) =
sum(predLDA==TestLabels)/size(TestLabels,1);
        TP = size(find(TestLabels == 0 & predLDA==0),1);
        TN = size(find(TestLabels == 1 & predLDA==1),1);
        FP = size(find(TestLabels == 1 & predLDA==0),1);
        FN = size(find(TestLabels == 0 & predLDA==1),1);
        sensitivity(ii, 3, LOO, FeatNum) = TP / (TP + FN);
        specificity(ii, 3, LOO, FeatNum) = TN / (TN + FP);
        precision(ii, 3, LOO, FeatNum) = TP / (TP + FP);
        f1_score(ii, 3, LOO, FeatNum) = 2 *TP/(2*TP+FP+FN);

        nTrees=200;
        B = TreeBagger(nTrees,NewDataTrain,TrainLabels, 'Method',
'classification');
        predTREE = B.predict(NewDataTest);
        predTREEC = str2double(predTREE);
        acc(ii,4,LOO,FeatNum) =
sum(predTREEC==TestLabels)/size(TestLabels,1);
        TP = size(find(TestLabels == 0 & predTREEC==0),1);
        TN = size(find(TestLabels == 1 & predTREEC==1),1);
        FP = size(find(TestLabels == 1 & predTREEC==0),1);
        FN = size(find(TestLabels == 0 & predTREEC==1),1);
        sensitivity(ii, 4, LOO, FeatNum) = TP / (TP + FN);
        specificity(ii, 4, LOO, FeatNum) = TN / (TN + FP);
        precision(ii, 4, LOO, FeatNum) = TP / (TP + FP);
        f1_score(ii, 4, LOO, FeatNum) = 2 *TP/(2*TP+FP+FN);

        c1 =
fitcsvm(NewDataTrain,TrainLabels,'KernelFunction','linear');
        predSVM = predict(c1,NewDataTest);
        acc(ii,5,LOO,FeatNum) =
sum(predSVM==TestLabels)/size(TestLabels,1);
        TP = size(find(TestLabels == 0 & predSVM==0),1);
        TN = size(find(TestLabels == 1 & predSVM==1),1);
        FP = size(find(TestLabels == 1 & predSVM==0),1);
        FN = size(find(TestLabels == 0 & predSVM==1),1);
        sensitivity(ii, 5, LOO, FeatNum) = TP / (TP + FN);
        specificity(ii, 5, LOO, FeatNum) = TN / (TN + FP);
        precision(ii, 5, LOO, FeatNum) = TP / (TP + FP);
        f1_score(ii, 5, LOO, FeatNum) = 2 *TP/(2*TP+FP+FN);

        % Update all bars
        frac1 = ii/size(FS_all_ranks,1)
        frac2 = FeatNum/ size(FS_all_ranks,3)
        frac3 = LOO/ size(cv,1)
        end
    end
end

```

Classification Optimization

```
% Classification with optimization
cv=CV;

FS_all_ranks =FS_all_ranks_dev;

for ii = 1:11%size(FS_all_ranks,1)
    for FeatNum = 1:500%size(FS_all_ranks,3)
        tempR = squeeze(FS_all_ranks(ii, :, :));
        Feats = [];
        temp2 = [];
        temp3 = [];
        temp2 = tempR(:, 1:FeatNum);
        for fold = 1:size(FS_all_ranks,2)
            temp3 = temp2(fold, :);
            Feats = horzcat(Feats,temp3);
        end
        [n,bin] = hist(Feats,unique(Feats));
        [~,idx] = sort(-n);
        b= n(idx); % count instances
        a=bin(idx); % corresponding values
        sortedFEATS = a(1:FeatNum);
        for L00 = 1:size(cv,1)

            %create train and test
            NewDataTest = features(cv(L00,:),sortedFEATS(1:FeatNum));
            TestLabels = labels(cv(L00,:),:);
            NewDataTrain = features(:,sortedFEATS(1:FeatNum));
            NewDataTrain(cv(L00,:),:) = [];
            TrainLabels = labels;
            TrainLabels(cv(L00,:),:) = [];

            frac1 = ii/size(FS_all_ranks,1)
            frac2 = FeatNum/ 200
            frac3 = L00/ size(cv,1)

            % Set up hyperparameter optimization options
            opts = struct(...
                'Optimizer', 'bayesopt', ...           % Use Bayesian optimization
                'ShowPlots', false, ...
                'Verbose', 0, ...
                'UseParallel', true, ...               % Use parallel computing if
available
                'MaxObjectiveEvaluations', 30);       % Adjust as needed

% KNN with Hyperparameter Optimization
KNNModel = fitcknn(NewDataTrain, TrainLabels, ...
    'OptimizeHyperparameters', 'auto', ...
    'HyperparameterOptimizationOptions', opts);

% Predict on the test data
pred = predict(KNNModel, NewDataTest);
% Compute accuracy
acc(ii,1,L00,FeatNum) = sum(pred == TestLabels) / length(TestLabels);
% Retrieve the best hyperparameters
bestKNNHyperparams =
KNNModel.HyperparameterOptimizationResults.XAtMinObjective;
% Optionally, store the best hyperparameters
```

```

bestKNN_NumNeighbors(ii, LOO, FeatNum) = bestKNNHyperparams.NumNeighbors;
bestKNN_Distance{ii, LOO, FeatNum} = bestKNNHyperparams.Distance;
    TP = size(find(TestLabels == 0 & pred==0),1);
    TN = size(find(TestLabels == 1 & pred==1),1);
    FP = size(find(TestLabels == 1 & pred==0),1);
    FN = size(find(TestLabels == 0 & pred==1),1);
    sensitivity(ii, 1, LOO, FeatNum) = TP / (TP + FN);
    specificity(ii, 1, LOO, FeatNum) = TN / (TN + FP);
    precision(ii, 1, LOO, FeatNum) = TP / (TP + FP);
    f1_score(ii, 1, LOO, FeatNum) = 2 *TP/(2*TP+FP+FN);

% SVM with RBF Kernel and Hyperparameter Optimization
SVMModel = fitcsvm(NewDataTrain, TrainLabels, ...
    'KernelFunction', 'rbf', ...
    'OptimizeHyperparameters', {'BoxConstraint',
'KernelScale'}, ...
    'HyperparameterOptimizationOptions', opts);
% Predict on the test data
predSVM = predict(SVMModel, NewDataTest);
% Compute accuracy
acc(ii,2,LOO,FeatNum) = sum(predSVM == TestLabels) / length(TestLabels);
    TP = size(find(TestLabels == 0 & predSVM==0),1);
    TN = size(find(TestLabels == 1 & predSVM==1),1);
    FP = size(find(TestLabels == 1 & predSVM==0),1);
    FN = size(find(TestLabels == 0 & predSVM==1),1);
    sensitivity(ii, 2, LOO, FeatNum) = TP / (TP + FN);
    specificity(ii, 2, LOO, FeatNum) = TN / (TN + FP);
    precision(ii, 2, LOO, FeatNum) = TP / (TP + FP);
    f1_score(ii, 2, LOO, FeatNum) = 2 *TP/(2*TP+FP+FN);

% Retrieve the best hyperparameters
bestSVMRBFHyperparams =
SVMModel.HyperparameterOptimizationResults.XAtMinObjective;
% Optionally, store the best hyperparameters
bestSVM_RBF_C(ii, LOO, FeatNum) = bestSVMRBFHyperparams.BoxConstraint;
bestSVM_RBF_KernelScale(ii, LOO, FeatNum) =
bestSVMRBFHyperparams.KernelScale;

% LDA with Hyperparameter Optimization
LDAModel = fitcdiscr(NewDataTrain, TrainLabels, ...
    'OptimizeHyperparameters', 'auto', ...
    'HyperparameterOptimizationOptions', opts);
% Predict on the test data
predLDA = predict(LDAModel, NewDataTest);
% Compute accuracy
acc(ii,3,LOO,FeatNum) = sum(predLDA == TestLabels) / length(TestLabels);
    TP = size(find(TestLabels == 0 & predLDA==0),1);
    TN = size(find(TestLabels == 1 & predLDA==1),1);
    FP = size(find(TestLabels == 1 & predLDA==0),1);
    FN = size(find(TestLabels == 0 & predLDA==1),1);
    sensitivity(ii, 3, LOO, FeatNum) = TP / (TP + FN);
    specificity(ii, 3, LOO, FeatNum) = TN / (TN + FP);
    precision(ii, 3, LOO, FeatNum) = TP / (TP + FP);
    f1_score(ii, 3, LOO, FeatNum) = 2 *TP/(2*TP+FP+FN);
% Retrieve the best hyperparameters
bestLDAModelHyperparams =
LDAModel.HyperparameterOptimizationResults.XAtMinObjective;
% Optionally, store the best hyperparameters
bestLDA_Delta(ii, LOO, FeatNum) = bestLDAModelHyperparams.Delta;
bestLDA_Gamma(ii, LOO, FeatNum) = bestLDAModelHyperparams.Gamma;

```

```

% Random Forest with Hyperparameter Optimization
RFmodel = fitcensemble(NewDataTrain, TrainLabels, ...
    'Method', 'Bag', ...
    'OptimizeHyperparameters', {'NumLearningCycles',
'MinLeafSize'}, ...
    'HyperparameterOptimizationOptions', opts);
% Predict on the test data
predRF = predict(RFmodel, NewDataTest);
% Compute accuracy
acc(ii,4,LOO,FeatNum) = sum(predRF == TestLabels) / length(TestLabels);
TP = size(find(TestLabels == 0 & predRF==0),1);
    TN = size(find(TestLabels == 1 & predRF==1),1);
    FP = size(find(TestLabels == 1 & predRF==0),1);
    FN = size(find(TestLabels == 0 & predRF==1),1);
    sensitivity(ii, 4, LOO, FeatNum) = TP / (TP + FN);
    specificity(ii, 4, LOO, FeatNum) = TN / (TN + FP);
    precision(ii, 4, LOO, FeatNum) = TP / (TP + FP);
    f1_score(ii, 4, LOO, FeatNum) = 2 *TP/(2*TP+FP+FN);
% Retrieve the best hyperparameters
bestRFHyperparams =
RFmodel.HyperparameterOptimizationResults.XAtMinObjective;
% Optionally, store the best hyperparameters
bestRF_NumTrees(ii, LOO, FeatNum) = bestRFHyperparams.NumLearningCycles;
bestRF_MinLeafSize(ii, LOO, FeatNum) = bestRFHyperparams.MinLeafSize;

% SVM with Linear Kernel and Hyperparameter Optimization
SVMLinearModel = fitsvm(NewDataTrain, TrainLabels, ...
    'KernelFunction', 'linear', ...
    'OptimizeHyperparameters', {'BoxConstraint'}, ...
    'HyperparameterOptimizationOptions', opts);
% Predict on the test data
predSVM = predict(SVMLinearModel, NewDataTest);
% Compute accuracy
acc(ii,5,LOO,FeatNum) = sum(predSVM == TestLabels) / length(TestLabels);
    TP = size(find(TestLabels == 0 & predSVM==0),1);
    TN = size(find(TestLabels == 1 & predSVM==1),1);
    FP = size(find(TestLabels == 1 & predSVM==0),1);
    FN = size(find(TestLabels == 0 & predSVM==1),1);
    sensitivity(ii, 5, LOO, FeatNum) = TP / (TP + FN);
    specificity(ii, 5, LOO, FeatNum) = TN / (TN + FP);
    precision(ii, 5, LOO, FeatNum) = TP / (TP + FP);
    f1_score(ii, 5, LOO, FeatNum) = 2 *TP/(2*TP+FP+FN);
% Retrieve the best hyperparameters
bestSVMLinearHyperparams =
SVMLinearModel.HyperparameterOptimizationResults.XAtMinObjective;
% Optionally, store the best hyperparameters
bestSVM_Linear_C(ii, LOO, FeatNum) =
bestSVMLinearHyperparams.BoxConstraint;
    end
end
end

```

Plotting Results and EEG Connections

```

%% Plot the results
ii=9;
a1=squeeze(acc(ii,:,:,:));
a2=squeeze(mean(a1,2));
a3=prod(a2,1);

```

```

[maxValue, idx] = max(a3(1:150));
FeatNum=idx

aa1=squeeze(specificity(9,:,:,:));
aa2=squeeze(sensitivity(9,:,:,:));
aa3=squeeze(f1_score(9,:,:,:));
aa4=squeeze(precision(9,:,:,:));
a2=squeeze(mean(a1,2));

plot(x, a2(1, :), 'r', 'LineWidth', 1.5); hold on;
plot(x, a2(2, :), 'g', 'LineWidth', 1.5);
plot(x, a2(3, :), 'b', 'LineWidth', 1.5);
plot(x, a2(4, :), 'c', 'LineWidth', 1.5);
plot(x, a2(5, :), 'm', 'LineWidth', 1.5);
hold off;

% Add legend for each row
legend({'KNN', 'SVM-RBF', 'LDA', 'RF', 'SVM-LIN'}, 'Location',
'northeast');

% Add labels and title
xlabel('Number of Features');
ylabel('Classifier Accuracy');
title('RFC-CBR-gaussian Optimized Accuracy');
grid on;

%a3=mean(a2,1);

plot(a3,'DisplayName','a3')
xlabel('Number of Features');
ylabel('Combined Accuracy');
title('RFC-CBR-gaussian Optimized Combined Accuracy');

%% Find EEG channels connections corresponding to the features
ii = 9;% feature selection
FeatNum = 19; % number of features
tempR = squeeze(FS_all_ranks(ii,:,:));
Feats = [];
temp2 = [];
temp3 = [];
temp2 = tempR(:,1:FeatNum);
for fold = 1:size(FS_all_ranks,2)
    temp3 = temp2(fold,:);
    Feats = horzcat(Feats,temp3);
end
[n,bin] = hist(Feats,unique(Feats));
[~,idx] = sort(-n);
b= n(idx); % count instances
a=bin(idx); % corresponding values
sortedFEATS = a(1:FeatNum);
band_names = {'Delta', 'Theta', 'Alpha', 'Beta', 'Gamma'};
num_bands = 5; % Number of frequency bands
num_subjects = 20; % Number of subjects
pli_matrix_size = 63; % Assume PLI is 63x63 matrix
num_features_per_band = (pli_matrix_size * (pli_matrix_size - 1)) / 2; %
Number of upper triangular elements
% Input: Element index in Feat_R
subject_idx = 1; % Subject index (row of Feat_R)
for i=1:length(sortedFEATS)
    element_idx = sortedFEATS(i); % Element index within Feat_R (column of
Feat_R)
% Find corresponding band

```

```

band = ceil(element_idx / num_features_per_band);
% Find position within the band
relative_idx = mod(element_idx - 1, num_features_per_band) + 1;
% Map the 1D index to row and column of the PLI matrix
% Create a lookup table for the upper triangular indices
[row_indices, col_indices] = find(triu(ones(pli_matrix_size), 1)');
row = row_indices(relative_idx);
col = col_indices(relative_idx);

channels ={'Fp1'   'AF7' 'AF3' 'F1'  'F3'  'F5'  'F7'  'FT7' 'FC5'
          'FC3' 'FC1' 'C1'  'C3'  'C5'  'T7'  'TP7' 'CP5' 'CP3' 'CP1'
          'P1'   'P3'  'P5'  'P7'  'P9'  'P07' 'P03' 'O1'  'Oz'  'POz'
          'Pz'   'CPz' 'Fpz' 'Fp2' 'AF8' 'AF4' 'Afz' 'Fz'  'F2'  'F4'
          'F6'   'F8'  'FT8' 'FC6' 'FC4' 'FC2' 'FCz' 'Cz'  'C2'  'C4'
          'C6'   'T8'  'TP8' 'CP6' 'CP4' 'CP2' 'P2'  'P4'  'P6'  'P8'
          'P10'  'P08' 'P04' 'O2'};
connections{1,i}=channels{row};
connections{2,i}=channels{col};
connections{3,i}= band_names{band};
end

```