Interactive 3D Visualization Framework for Machine Learning-Based Network Intrusion Detection Systems

by

Jamal Hossain
ID: 2038520137

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science and Engineering

Department of Computer Science and Engineering

University of Rajshahi

November, 2025

BOARD OF EXAMINERS

Supervisor

Subrata Pramanik
Associate Professor,
Department of Computer Science and Engineering
University of Rajshahi

Signature

Declaration

I hereby declare that the work presented in this thesis is the outcome of our own research, conducted under the supervision of Subrata Pramanik, Associate Professor,Department of Computer Science and Engineering ,University of Rajshahi. I further declare that this thesis, in whole or in part, has not been submitted to any other institution or organization for the award of any degree, diploma, or certificate, and that all sources of information used in this work have been properly acknowledged.

Student's Full Name & Signature:

Jamal Hossain

Acknowledgement

All praise is due to Almighty Allah, whose mercy and blessings have guided us through every stage of this research journey with clarity and purpose.

I would like to express our deepest gratitude to our supervisor, Subrata Pramanik, Associate Profes- sor,Department of Computer Science and Engineering ,University of Rajshahi, for his invaluable guidance, continuous encouragement, and insightful feedback throughout the course of this work. His mentorship has been instrumental in shaping the direction and quality of our research.

I also extend our sincere appreciation to the faculty members of the Department of Computer Science and Engineering for their academic support and constructive suggestions, which have significantly contributed to our learning and progress.

Finally, I am deeply grateful to our families for their unwavering encouragement, patience, and moral support, which sustained us throughout the challenges of this academic journey.

Abstract

In the modern digital era, cyber-attacks have become increasingly frequent and complex, posing serious threats to computer networks and data security. To mitigate these risks, it is essential to develop robust and efficient systems capable of detecting and responding to such attacks promptly. One promising approach is the integration of Machine Learning (ML) into Network Intrusion Detection Systems (NIDS) to automatically identify abnormal network behavior. However, despite the success of ML models in detecting threats, they often misclassify certain types of traffic, and understanding the underlying causes of these errors remains a significant challenge. This thesis presents a three-dimensional interactive visualization framework designed to support the analysis and interpretation of such misclassifications. The proposed system visually illustrates how different categories of network traffic are classified, highlighting areas where they may be incorrectly separated or mixed. This approach enables researchers and practitioners to gain deeper insights into model performance and recognize recurring misclassification patterns. Two benchmark datasets, NSL-KDD and UNSW-NB15, are used to evaluate the effectiveness of the framework. Standard evaluation metrics, including accuracy, precision, recall, F1-score, and ROC-AUC, are employed to quantitatively assess the performance of the ML models integrated within the system. The primary objective of this research is to enhance the interpretability of ML model results and provide meaningful insights for improving the design and performance of future intrusion detection systems.

Keywords: Cybersecurity, Network Intrusion Detection, Machine Learning, Misclassification Analysis, 3D Visualization, PCA, LDA, NSL-KDD, UNSW-NB15, Decision Boundary,Interactive System.

Table of Contents

List of Figures

List of Tables

CHAPTER I

INTRODUCTION

1.1 Background

The increasing frequency, complexity, and impact of cyber attacks have made network security a critical concern in today's digital infrastructure. Among the various defense mechanisms, Network Intrusion Detection Systems (NIDS) play a vital role in identifying unauthorized or malicious activity within network traffic. In recent years, machine learning (ML) techniques have been widely adopted in NIDS, offering the ability to automatically detect anomalies and attack patterns through the analysis of large volumes of network data.

Despite their potential, ML based intrusion detection models face persistent challenges, particularly in relation to misclassification. Certain traffic instances, such as minority class attacks or previously unseen behaviors, are often incorrectly labeled due to overlapping feature distributions, class imbalance, or limited generalization capabilities. These misclassifications can significantly reduce detection performance and increase false alarm rates.

A further challenge lies in the limited interpretability of many ML models. Complex algorithms such as Support Vector Machines and neural networks often operate as black boxes, making it difficult to understand the reasoning behind specific predictions, especially when errors occur. Traditional evaluation metrics like accuracy, precision, F1 score and recall provide numerical summaries but offer little insight into where and why models fail.[3]

To overcome this limitation, visual analytics has emerged as a promising approach to enhance model transparency. By applying dimensionality reduction techniques such as Principal Component Analysis (PCA) and visualizing high-dimensional network data in a two or three-dimensional space, it becomes possible to observe the structure of data, explore classification boundaries, and identify regions that are prone to misclassification. These visual representations support an intuitive understanding of the behavior of the model and provide a basis for a more informed refinement of the model.[10]

This thesis proposes the development of a three-dimensional interactive visualization framework for analyzing misclassification in machine learning models used for network intrusion detection. By projecting network traffic data and classifier outcomes

into a visually interpretable space, the framework aims to reveal data patterns, support error analysis, and contribute to the advancement of more explainable and effective intrusion detection models.

1.2 Motivation

Network Intrusion Detection Systems (NIDS) play a critical role in identifying malicious activities within network traffic. Although machine learning has significantly improved detection accuracy, most existing models behave like black boxes, providing predictions without explaining how those decisions are made. This lack of interpretability makes it difficult to understand the causes of misclassification and limits the ability to improve the model effectively.[1]

Current approaches mainly rely on statistical metrics such as accuracy, precision, F1 score and recall, which summarize performance but do not reveal the reasoning behind individual predictions or errors. This highlights a key gap: the absence of tools that allow analysts to visually explore and understand the decision boundaries and behavior of ML models.

This research is motivated by the need to bridge that gap through the development of a three-dimensional interactive visualization framework. By enabling users to visually interpret how machine learning models classify different types of network traffic, the system can help identify areas where misclassification occurs and explain why those errors happen. This visual insight supports more informed model refinement, enhances transparency, and contributes to building more reliable and effective intrusion detection systems.[5]

1.3 Problem Statement

Machine learning has become an essential component of modern Network Intrusion Detection Systems (NIDS), enabling automatic detection of malicious activities within network traffic. However, a significant problem in these systems is the occurrence of misclassification, where normal traffic may be incorrectly labeled as malicious, or harmful traffic may be mistakenly identified as safe. These errors can lead to serious consequences, such as system compromise or unnecessary alerts, both of which reduce the reliability of the system.

Moreover, most machine learning models used in NIDS act as black boxes, providing predictions without explaining the reasoning behind them. Traditional evaluation methods, such as accuracy scores and confusion matrices, offer only numerical summaries and do not help analysts understand why the model made a particular

decision. As a result, it becomes difficult to identify the causes of misclassification or to improve the model in a meaningful way.[11]

The lack of interpretability and the inability to visually analyze how machine learning models behave in intrusion detection tasks represent a critical gap in current research and practice. There is a clear need for a solution that allows users to explore and understand model decisions, particularly in the context of misclassified data.[12]

To address this gap, this research proposes the development of a three-dimensional interactive visualization framework.The goal is to help users observe decision boundaries, identify regions where misclassification occurs, and gain visual insight into the internal behavior of machine learnng models used in NIDS. This will support better analysis, improve model transparency, and contribute to building more reliable and interpretable intrusion detection systems.

1.4 Objectives

The primary objective of this research is threefold:

1. To Develop a 3D interactive visualization framework for analyzing machine learning misclassifications in network intrusion detection models, providing a clear visual representation of decision boundaries and misclassified regions.[5]

2. To Assess the effectiveness of dimensionality reduction techniques, such as Principal Component Analysis (PCA)[10], Linear Discriminant Analysis (LDA) in transforming high-dimensional network traffic data into a form that is suitable for interactive exploration and visualization.

3. To enhance the interpretability and transparency of machine learning models in NIDS, enabling users to interactively explore the decision-making processes and identify areas for model refinement and error correction.

1.5 Unfamiliarity of the Problem

Despite advancements in machine learning-based Network Intrusion Detection Systems (NIDS), a major challenge persists: the lack of interpretability and transparency in the decision-making process. While these algorithms show promising results in detecting intrusions, they are often criticized as black boxes, providing predictions without clear explanations. This issue becomes particularly problematic when the system misclassifies network traffic, as analysts struggle to understand the

causes of errors or improve the model. Traditional performance metrics like accuracy and precision fail to offer insights into the internal workings of the model and overlook regions in the decision space where misclassifications are likely to occur. The absence of visual and interpretive tools hinders targeted model refinement.[3]

This research addresses this gap by developing an interactive 3D visualization framework, enabling more intuitive and comprehensive analysis of machine learning misclassifications in NIDS. By visually exploring the decision-making process, this framework enhances interpretability and provides actionable insights to improve model accuracy and reliability[6].

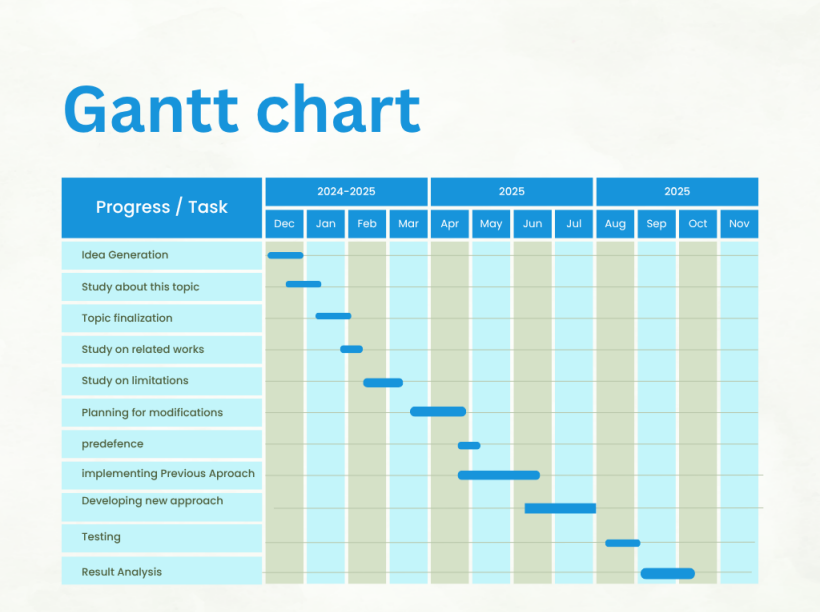
1.6 Contribution

This thesis presents an NIDS framework combining PCA + LDA, ensemble learning, and interactive 3D visualization. PCA + LDA improves class separability, while the 3D visualization enhances interpretability. Ensemble methods such as Random Forest and XGBoost reduce overfitting. Evaluation on the UNSW-NB15 and NSL-KDD datasets demonstrates improved performance and effective handling of class imbalance, offering a comprehensive model evaluation through both statistical metrics and visual analytics.

In this thesis,
Jamal Hossain focused on the literature review, dataset analysis, data preprocessing, pipeline implementation, evaluation metrics, interactive 3D visualization framework, and thesis writing.
This work enhances NIDS performance, improving interpretability, robustness, and applicability in real-world cybersecurity contexts.

1.7 Project Planning

The thesis structure was organized using a Gantt chart, providing an overview of research activities and milestone progression (Figure 1.1). The preliminary phase involved idea development and an extensive literature review to establish a solid conceptual foundation, which required additional time to synthesize findings, identify gaps, and highlight areas for improvement.



The next phase focused on evaluating prior studies to refine the framework for improved precision and robustness. The framework was then developed, implemented, and validated according to the Gantt chart timeline. With all objectives achieved and results verified, the thesis is now ready for academic defense.

1.8 Organization

This thesis is structured into six sections, each addressing a specific component of the research. Section 1 introduces the study by presenting the background, motivation, problem statement, objectives, and scope of the work. Section 2 provides a comprehensive review of the existing literature on network intrusion detection and the application of machine learning techniques in this domain. It also examines previous research employing three-dimensional visualization approaches and identifies the key research gaps that form the foundation of this study. Section 3 outlines the tools, technologies, and datasets utilized in the research, along with details of the experimental setup and implementation environment. Section 4 elaborates on the research methodology, including the system architecture, data preprocessing procedures, model development process, and the design of the proposed visualization framework for analyzing misclassifications. Section 5 presents and interprets the experimental results, emphasizing misclassification analysis and the evaluation metrics used to assess the performance and reliability of the proposed system. Finally, Section 6 concludes the thesis by summarizing the principal findings, highlighting the study's contributions, and suggesting potential directions for future research aimed at enhancing the interpretability and effectiveness of machine-learning-based intrusion detection systems.

CHAPTER II

RELATED WORK

Network Intrusion Detection Systems (NIDS) serve as a vital defense layer against the increasing number and complexity of cyber-attacks. Traditional signature-based systems can effectively detect known threats but often fail to recognize new or unknown attacks. To overcome these limitations, researchers have increasingly adopted Machine Learning (ML) approaches capable of automatically identifying abnormal network behavior through data-driven analysis. Among these approaches, dimensionality reduction techniques such as Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) are widely used to enhance detection accuracy, reduce redundant features, and improve computational efficiency. While PCA focuses on capturing the most significant variations in the data, LDA emphasizes distinguishing between classes by maximizing class separability.

Recent research has combined these methods with feature selection techniques to refine input variables and improve model generalization. In parallel, visualization methods have gained attention for making ML-based NIDS more interpretable. Two- and three-dimensional (3D) visualizations allow analysts to better understand model decisions, identify misclassifications, and explore relationships among different traffic categories. Therefore, integrating ML algorithms with dimensionality reduction, feature selection, and visualization provides a strong foundation for developing more accurate, efficient, and interpretable intrusion detection frameworks.

Bulavas [7] proposed a network intrusion detection method utilizing data visualization techniques. The method employs dimensionality reduction and various visualization techniques to identify anomalies and potential intrusions in network traffic. The approach utilizes Principal Component Analysis (PCA) to project high-dimensional data onto fewer dimensions, making it easier to identify distinct attack patterns. The technique also integrates machine learning, particularly Decision Trees, to improve intrusion detection accuracy. This method can effectively classify intrusions such as DoS, U2R, and R2L with high accuracy. It also accommodates new attack patterns by dynamically adapting to changes in network conditions.

Camacho et al. [4] proposed a multivariate statistical network monitoring technique for anomaly detection, specifically using Principal Component Analysis (PCA) to monitor network traffic. The proposed anomaly detection scheme for network traffic operates through multivariate statistical process control (MSPC), utilizing PCA to identify both common and special causes of variation. This scheme aims to enhance the accuracy of anomaly detection by leveraging multiple network variables rather than focusing on single-dimensional data.

Liu, Shixia et al. [5] proposed a technique for analyzing machine learning models from a visual analytics perspective. The proposed method integrates various visualization techniques to enhance the interpretability of machine learning models. It effectively maps model behavior through interactive visual interfaces, making it easier to understand and analyze complex models. The approach utilizes dimensionality reduction and data clustering, allowing for intuitive exploration of data features and model predictions. It also dynamically adjusts its operations to accommodate different datasets, ensuring efficient processing while maintaining interpretability.

Ruan et al. [6] proposed a novel approach to visualizing big data security, specifically in the context of the KDD99 data set. The proposed method combines a unique sampling technique with visualization algorithms to improve the understanding of cybersecurity data. This technique addresses challenges such as data redundancy and bias by reducing the complexity of the data set and making it more manageable for analysis. The scheme allows for clearer identification of attack patterns and "normal" clusters, enhancing the ability of intrusion detection systems (IDS) to recognize and categorize various types of cyber attacks. The approach also accommodates the inherent issues of big data, including volume, variety, and velocity, while maintaining visualization accuracy and efficiency.

Tauscher et al. [12] proposed a data-driven approach to network intrusion detection using machine learning models. The proposed method performs a comprehensive analysis on the NSL-KDD dataset, utilizing both supervised and unsupervised learning techniques for anomaly detection and classification of various attack types. The system uses a two-stage hierarchy approach, where normal behaviors are separated from anomalies, and then further classified into specific attack types such as DoS, Probe, R2L, and U2R. It also addresses the issue of data imbalance through the SVM-SMOTE oversampling technique. The method demonstrates the effectiveness of unsupervised representation learning models and highlights the advantages of combining multiple learning algorithms in detecting and classifying network intrusions.

Singh et al. [11] proposed a hybrid intrusion detection system using Support Vector Machines (SVM) and k-Nearest Neighbor (k-NN). The proposed method functions in a two-step process: first, SVM is used to classify the data into categories, followed by k-NN to further classify uncertain records. This hybrid approach helps to improve accuracy by addressing the uncertainty in the initial classification step. The system is evaluated on the NSL-KDD dataset, and the results show that the proposed model compares favorably with other recent intrusion detection approaches.

Yelizarov and Gamayunov [1] proposed a visualization technique to effectively display complex network attacks, such as multi-step and DDoS attacks. The method enables network administrators to view both ongoing simple events and the full context of complex attacks, including their severity, duration, and interrelations. The system provides a 3D visualization space, where attacks are represented as glyphs, with color mapping to indicate attack types and glyph height representing severity. This approach enhances the administrator's ability to perceive attack patterns and relationships between events, providing a clearer and more intuitive understanding of ongoing threats.

Kamarudin et al. [9] proposed a hybrid feature selection model for intrusion detec-

tion systems. The proposed model combines filter-based and wrapper-based feature selection methods to improve intrusion detection accuracy. It uses Correlation Feature Selection (CFS) along with three different search techniques best-first Search, greedy stepwise, and genetic algorithm to identify the optimal feature subset. The method effectively reduces the number of features while maintaining high detection accuracy. The model was tested on the KDD99 and DARPA 1999 datasets using a Random Forest classifier, showing promising results in terms of detection rate and accuracy.

Almomani et al. [13] proposed a multi-stage machine learning framework for intrusion detection in IoT environments. The proposed approach enhances detection performance by integrating optimized feature selection and ensemble learning techniques. It effectively detects various types of attacks with high accuracy and low false alarm rates across multiple benchmark datasets. The system demonstrates robustness, adaptability, and efficiency, making it suitable for securing dynamic IoT scenarios.

Ahmed, A., and Kalita, J. [8] proposed an ensemble-based intrusion detection framework designed to enhance detection accuracy in complex network environments. The framework leverages multiple machine learning classifiers, including Decision Trees, Random Forest, and AdaBoost, combined with feature selection techniques to optimize performance. It emphasizes the importance of reducing false positives and improving detection precision by analyzing the significance of selected features across different classifiers.

Buczak and Guven [3] presented a comprehensive survey of machine learning techniques for intrusion detection systems (IDSs). The study categorizes IDSs based on supervised, unsupervised, and hybrid learning approaches and analyzes their applicability, strengths, and limitations. It emphasizes the importance of feature selection, model interpretability, and real-time performance in IDS development.

Zong et al. [10] proposed an interactive three-dimensional (3D) visualization approach for network intrusion detection data to enhance the interpretability of machine learning models. The approach enables analysts to better understand how different types of network traffic are classified by illustrating their geometric relationships in 3D space. Using Principal Component Analysis (PCA), the authors reduced high-dimensional data from the UNSW-NB15 and NSL-KDD datasets into three dimensions, revealing distinct clusters, class boundaries, and instances of misclassification. They also noted that while PCA effectively preserves the overall data structure, some information loss occurs during dimensionality reduction, highlighting the need for further research into alternative visualization techniques.

Sangkatsanee et al. [2] proposed a real-time intrusion detection system utilizing the Decision Tree algorithm to classify network data as normal or attack data. By applying information gain to select the most relevant features, the model achieved 98% detection accuracy with minimal processing time.

2.1 Dimensionality Reduction and Machine Learning in NIDS

Dimensionality reduction simplifies high-dimensional traffic features and often boosts IDS performance. PCA is commonly used to highlight attack patterns while reducing noise and redundancy. Studies such as [4], [5], [6], [7], [9], [12] combine PCA (or feature selection) with ML to improve accuracy, handle imbalance, and support interpretability.

Bulavas [7] used PCA for dimensionality reduction to enhance attack detection accuracy, including DoS, U2R, and R2L.

Liu et al. [5] integrated visualization and dimensionality reduction to improve machine learning model interpretability and detection accuracy. Ruan et al. [6] used PCA and clustering for reducing data complexity and improving attack pattern detection. Combining machine learning and dimensionality reduction provides more efficient and accurate intrusion detection by focusing on the most relevant features.

Tauscher et al. [12] combined unsupervised learning and PCA to detect anomalies and handle class imbalance.

Kamarudin et al. [9] used a hybrid feature selection approach with dimensionality reduction, improving detection rates on datasets like KDD99

Camacho, J., P´erez-Villegas, A., Garc´ıa-Teodoro, P., Maci´a-Fern´andez, G. [4]PCA-based multivariate statistical network monitoring for anomaly de tection. Computers Security, 59, 118-137

2.2 Network Intrusion Visualization

Network intrusion visualization helps in identifying complex attack patterns and improving detection accuracy by transforming raw data into understandable formats. Several studies have explored the combination of visualization techniques and intrusion detection systems (IDS).

Ruan et al. [6] developed a 3D visualization approach using PCA to project network intrusion data from the UNSW-NB15 and NSL-KDD datasets, revealing class clusters and misclassifications while noting minor information loss during dimensionality reduction.

Zong et al. [10] developed an interactive 3D visualization approach using machine learning to enhance IDS training with synthetic data generated by GANs.

2.3 Network Intrusion Detection Datasets

Network intrusion detection datasets are essential for training and evaluating intrusion detection systems (IDS). These datasets provide labeled examples of normal

and malicious network traffic, helping to test the performance of different models.

Kamarudin et al. [9] evaluated their hybrid feature selection model using KDD99 and DARPA 1999 datasets, which are widely used for NIDS research.

Zong et al. [10] used the NSL-KDD dataset to improve detection accuracy, especially for detecting minority class attacks.

Tauscher et al. [12] utilized the NSL-KDD dataset for testing ML models in NIDS, highlighting the importance of datasets in evaluating IDS performance.

Buczak and Guven [3] discussed datasets including KDD99, NSL-KDD, and DARPA, emphasizing challenges like imbalance and evolving attack patterns.

Table 2.1: Comparison of related works with proposed work

Works	**Datasets**	**Feature Reduction**	**Model**	**Visualization**
Zong et al. [10]	NSL-KDD, UNSW-NB15	PCA	SVM	3D
Bulavas [7]	NSL-KDD	PCA	DT	2D
Tauscher et al. [12]	NSL-KDD	N/A	DT, RF	2D
Ruan et al. [6]	KDD99	PCA, MDS	SVM	2D
Sangkatsanee et al. [2]	RLD09	N/A	DT, NB	N/A
Liu et al. [5]	KDD99	t-SNE	CNN, SVM	2D
Proposed Work	NSL-KDD, UNSW-NB15	PCA, LDA	AdaBoost, Bagging, CatBoost, Gradient Boosting, KNN, RF, SVM, XGBoost, Ensemble	3D

CHAPTER III

REQUIRED TOOLS

This chapter outlines the essential tools and software required for the development of the 3D Interactive Visualization Framework for analyzing machine learning misclassification in network intrusion detection models. The framework integrates machine learning, dimensionality reduction techniques, and 3D visualization to provide insights into misclassification and decision boundaries in network intrusion detection systems (NIDS).

3.1 Programming Languages and Development Environments

The core programming language used in the framework is Python, chosen for its extensive ecosystem of libraries that support machine learning, data analysis, and visualization. The following tools and libraries are critical to the development process:

1. Python : The primary language for model development, data processing, and algorithm implementation.

2. Scikit-learn : A core library used for implementing machine learning algorithms such as Support Vector Machines (SVM), essential for classifying network traffic.

3. NumPy and Pandas: Libraries used for efficient data manipulation and preprocessing of large-scale datasets.

4. Matplotlib : Libraries for generating 2D and 3D plots, aiding in the visualization of data distributions, model outputs, and performance metrics.

5. Jupyter Notebook : Jupyter Notebook is an open-source interactive platform that allows users to write and execute code, visualize data, and add explanations in one place. It is widely used in data analysis, machine learning, and academic research for combining code, output, and documentation seamlessly.

3.2 Machine Learning and Dimensionality Reduction

The framework utilizes machine learning models to classify network traffic as either normal or malicious. Dimensionality reduction techniques are employed to simplify high dimensional data while preserving essential features necessary for effective classification and visualization.

1. Dimensionality Reduction Techniques: Principal Component Analysis (PCA) is used to reduce the number of features while retaining those most significant for classification and visualization. It transforms high-dimensional data into lower-dimensional forms (e.g., 2D or 3D), making the data easier to visualize without compromising its integrity.

2. Machine learning Classifier: ML is used to classify data based on features extracted after dimensionality reduction. It distinguishes between normal and attack traffic and helps define the decision boundaries for 3D visualization.

3.3 Visualization Tools

Visualization is a core component of the proposed framework. It enables a spatial understanding of misclassification patterns and decision regions in the dataset.

1. Plotly: Utilized for creating an interactive 3D visualization environment with real-time WebGL rendering, allowing users to dynamically explore network traffic patterns and classifier decision boundaries directly within a web-based analytical interface.

2. Unity 3D: Utilized for creating an interactive 3D environment with real-time rendering and spatial navigation, enabling users to visually explore network traffic patterns and classifier decision spaces within a simulation-based virtual platform.

3. Blender: Used when advanced 3D modeling or animations are required to enhance the visual interface.

4. Matplotlib: Assist in generating 2D projections of 3D data, which help illustrate decision spaces and misclassification patterns effectively.

In this thesis, Plotly was used for interactive 3D visualization, providing a web based, GPU accelerated, and CPU efficient framework for analyzing high dimensional intrusion data. Unlike traditional Unity 3D or static scatter based methods, Plotly leverages WebGL rendering to deliver smooth, real time interaction allowing

users to rotate, zoom, and inspect data points dynamically with minimal computational cost. It also supports custom tooltips, multi layered views, and Python integration, enhancing both interpretability and accessibility. Overall, the Plotly based visualization offers a scalable, resource efficient, and analytically precise approach for examining class boundaries, cluster overlaps, and misclassification regions in complex network traffic data.

3.4 Data Preprocessing and Dataset Utilization

The framework leverages publicly available and widely accepted NIDS datasets such as NSL-KDD and UNSW-NB15 for training and evaluating the machine learning models.

1. Data Preprocessing: Conducted using Pandas and NumPy for tasks like handling missing values, feature normalization, and encoding categorical variables.

2. Feature Engineering: Involves removing redundant or irrelevant features using techniques such as PCA and information gain, thereby improving model focus and performance.

3. Datasets: NSL-KDD and UNSW-NB15 include labeled samples of both attack and normal traffic, covering various attack categories such as Denial of Service (DoS), Remote to Local (R2L), User to Root (U2R), and probing.

3.5 Hardware Requirements

To support machine learning computations and real-time 3D rendering, the following hardware configuration is recommended:

1. CPU: Intel Core i5 (quad-core or higher).

2. RAM: A minimum of 4 GB is required; 8 GB or more is recommended for large dataset handling.

3. Graphics Card: NVIDIA GeForce GTX series (or higher) is suggested for optimal 3D rendering and visualization performance.

3.6 Additional Tools

1. TensorFlow or PyTorch (optional): These deep learning libraries can be integrated if the framework is extended to incorporate deep learning based intrusion detection models.

2. Jupyter Notebooks: Provide an interactive environment for rapid experimentation, prototyping, and testing of machine learning models and visualizations.

3. Google Colab: Cloud-based Jupyter environment with free GPU/TPU support for collaborative, setup free development.

4. LaTeX: Used for creating well-formatted technical documents with mathematical notation and structured content.

This chapter presented the essential tools, libraries, and hardware necessary to develop the 3D interactive visualization framework. By combining machine learning, dimensionality reduction, and advanced visualization technologies, the system provides a comprehensive environment for understanding classification performance and identifying misclassification patterns in network intrusion detection systems. The interactive visualization component enhances the interpretability of machine learning models, making the framework both informative and user-friendly.

CHAPTER IV

METHODOLOGY

This section outlines the design and development of an interactive 3D visualization framework for analyzing machine learning (ML) misclassification in network intrusion detection systems (NIDS). The approach integrates machine learning models with 3D visualization to provide a better understanding of misclassification occurrences and the performance of intrusion detection models.

4.1 Overview

The proposed approach uses 3D visualization to display the geometric relationships between various categories of network traffic. By visualizing the machine learning decision spaces, the framework allows for interactive exploration of decision boundaries, highlighting instances where misclassification occurs. This enables a more intuitive understanding of ML model performance and helps address the challenges of interpreting complex detection results.

4.2 Datasets Analysis

For the development and evaluation of the proposed framework, network traffic datasets are chosen to represent different types of intrusion attacks and normal traffic. These datasets are crucial for training machine learning models and evaluating misclassification performance.Table 4.1 summarizes the key characteristics of each datasets.

UNSW-NB15: Released by the University of New South Wales; widely used for NIDS. It contains attacks such as DoS, Probing, R2L, U2R, and normal traffic.Both categorical and numerical features are present.

Table 4.1: Datasets , total samples, and feature counts

Dataset[14],[15]	**Total Samples**	**Number of Features**
UNSW - NB15 (Train)	82,332	45
UNSW - NB15 (Test)	175,342	
NSL - KDD (Train)	125,973	42
NSL - KDD (Test)	22,544	

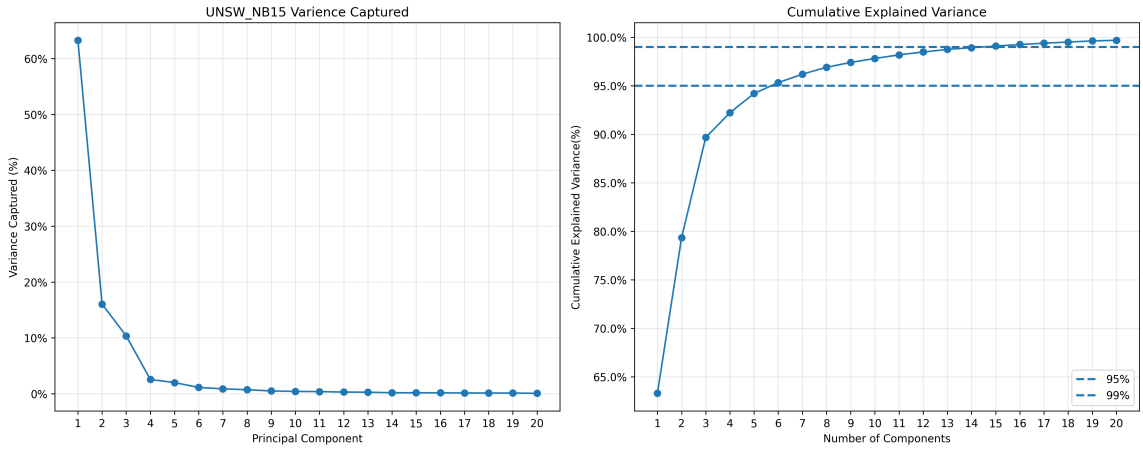


Figure 4.1: Variance of UNSW-NB15 dataset

Principal Component Analysis (PCA) was applied to the UNSW-NB15 dataset to uncover the main patterns of variance and understand how well different classes can be separated. The first principal component captured 63% of the total variance, while the second and third components explained 17% and 10%, respectively. Together, the first six components accounted for more than 95% of the overall variance, showing that the dataset has a moderate level of intrinsic dimensionality and that many of its features are strongly correlated.

While PCA helped simplify the dataset and highlight its key structures, some attack categories especially Normal, Generic, and Exploits showed overlapping regions in the reduced feature space. These overlaps often caused misclassifications, as the boundaries between classes became less distinct. To explore these overlaps and visualize the classifier's decision boundaries more clearly, a 3D PCA plot using the first three components was created. This visualization made it easier to spot ambiguous regions and provided valuable insight for refining both the feature extraction and classification stages, ultimately improving the accuracy of intrusion detection.

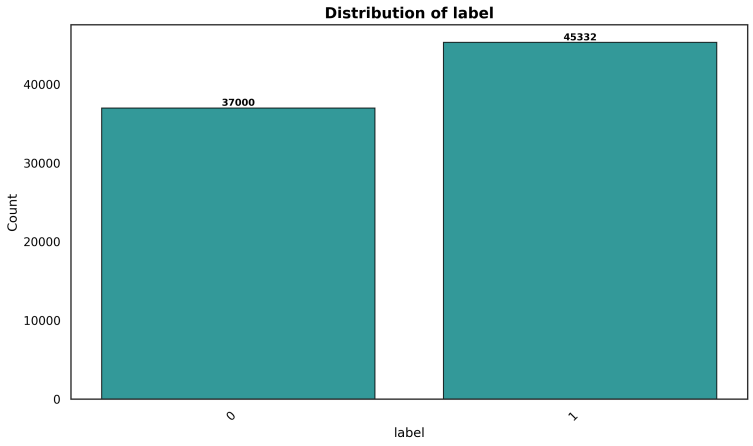


Figure 4.2: Distribution level of UNSW-NB15 dataset

The binary label distribution shows a moderate class imbalance, with 45,332 attack

samples (label = 1) and 37,000 normal samples (label = 0). While this imbalance is not severe, the larger number of attack instances can still influence how the model learns, making it slightly more inclined to predict attack behavior.

NSL-KDD: Improved over KDD'99; 42 features and five categories (DoS, R2L, U2R, Probing, Normal). Redundant records removed for a more balanced distribution.

These datasets provide a diverse set of traffic and attack types needed to evaluate the framework. Training and testing of the ML models rely on these datasets to assess misclassification and the effectiveness of the 3D visualization.

Principal Component Analysis (PCA) was applied to the NSL-KDD dataset to identify the dominant directions of variance and evaluate class separability. The first principal component captured 68% of the total variance, followed by 12% and 5% for the second and third components, respectively. More than 95% of the total variance was explained by the first five components, confirming the dataset's low intrinsic dimensionality and high inter-feature correlation.

Despite this compact representation, certain classes (e.g., DoS, Normal, Probe) exhibited overlapping feature distributions, leading to misclassifications within the reduced subspace. To analyze these overlaps and the classifier's decision boundaries more effectively, a 3D PCA visualization based on the first three components was employed. This approach provided a clearer view of ambiguous regions and informed subsequent model refinements to improve class discrimination.

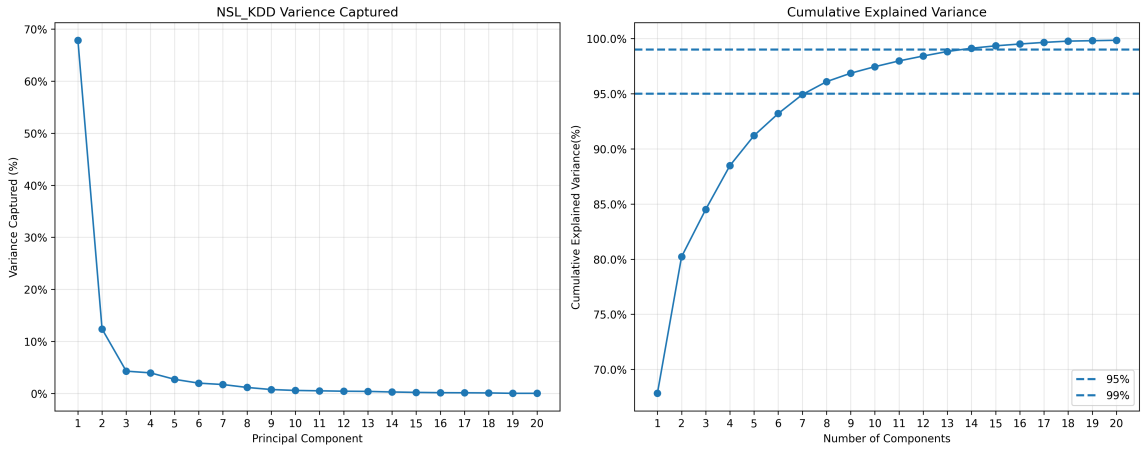


Figure 4.3: Variance of NSL-KDD dataset

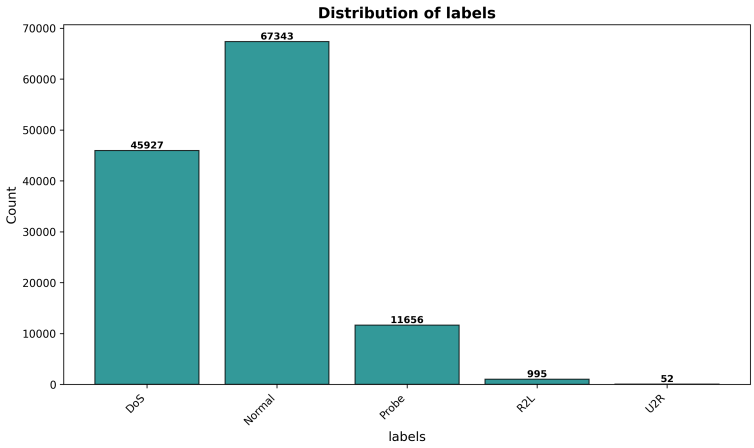


Figure 4.4: Distribution level of NSL-KDD dataset

The NSL-KDD dataset exhibits a pronounced class imbalance. The Normal class dominates with 67,343 samples, followed by DoS with 45,927, and Probe with 11,656 instances. In contrast, the minority classes R2L and U2R contain only 995 and 52 samples, respectively.

This imbalance significantly affects model learning, as classifiers tend to prioritize majority patterns while overlooking rare attack behaviors. Consequently, models may achieve high overall accuracy but perform poorly in detecting low-frequency intrusions such as R2L and U2R. To mitigate this issue, resampling strategies (e.g., SMOTE, SMOTE+Tomek) were integrated into the preprocessing pipeline, and class-balanced evaluation metrics (e.g., macro F1-score, recall) were employed. These steps ensured that the learning process remained sensitive to minority classes, improving both detection robustness and generalization capability across all attack categories.

Such imbalance can affect the decision boundary, increasing the chance of false negatives and reducing the model's ability to accurately identify normal traffic. To mitigate this, class-balanced evaluation metrics such as macro-averaged F1-score and recall were used to ensure fair performance across both classes. These measures help maintain consistent detection reliability and improve the model's overall robustness in distinguishing between normal and malicious network activities.

4.3 Pipeline of the Studied Methodology

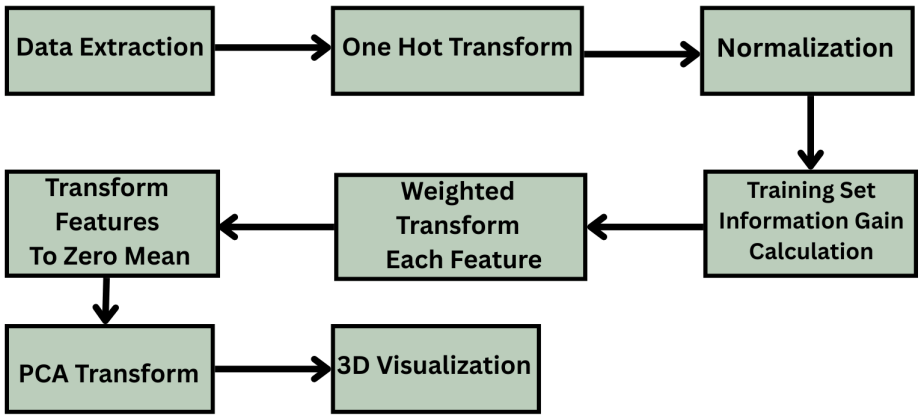


Figure 4.5: Studied methodology [10].

1. Data Extraction. Network traffic is collected from sources such as UNSW-NB15 and NSL-KDD. Features include packet size, flow type, and attack type.

2. One-Hot Transform. Categorical features are converted to binary indicators:

$$x_i = \begin{cases} 1 & \text{if the feature equals the category,} \\ 0 & \text{otherwise.} \end{cases}$$

3. Normalization. Numerical features are scaled to $[0, 1]$:

$$X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}}.$$

4. Training Set Information Gain. For feature A,

$$IG(S, A) = E(S) - \sum_{v \in A} \frac{|S_v|}{|S|} E(S_v), \qquad E(S) = -\sum_{i=1}^{n} p_i \log_2 p_i.$$

5. Weighted Transformation. Each feature receives a weight c_i according to importance (from feature selection or domain knowledge).

6. Zero-Mean Centering.

$$X' = X - \mu.$$

Table 4.2: (x, y, z) components of a basis vector in 3D space.

Vector	**Components (x,y,z)**
$\hat{v}_1$	(1, 0, 0)
$\hat{v}_2$	(0, 1, 0)
$\hat{v}_3$	(0, 0, 1)

7. PCA Transform.

$$X' = X\,W,$$

where W contains the eigenvectors (principal components).

8. 3D Visualization. Points are mapped into 3D:

$$\text{position} = \sum_{i=1}^{3} c_i\,\hat{v}_i\,s,$$

where $\hat{v}_1 = (1, 0, 0)$, $\hat{v}_2 = (0, 1, 0)$, $\hat{v}_3 = (0, 0, 1)$, and s is a scale in the Table 4.2

4.4 Decision Space

A voxel-based approach was employed to visualize the machine learning (ML) decision space for various network traffic categories. This 3D visualization represents regions where the ML model classifies traffic. For instance, network instances that fall within the Denial-of-Service (DoS) decision space are classified as DoS attacks, while those outside are misclassified, allowing users to assess the model's classification accuracy and identify potential misclassifications.

The visualization was evaluated using eight base models (AdaBoost, Bagging, CatBoost, Gradient Boosting, K-Nearest Neighbors, Random Forest, Support Vector Machine, XGBoost), along with two ensemble models (Voting Classifier, Stacking Classifier).

To optimize the classification process, the first 13 principal components were selected, accounting for 96.3% of the variance in the UNSW-NB15 dataset and 98.7% in the NSL-KDD dataset, with the remaining components contributing negligibly. This reduction in features helps minimize computational costs without sacrificing model accuracy.

In the 3D visualization, the decision space was divided into 150 voxels along each axis, resulting in over 3 million voxels. Each voxel's center was extended to 13D by initializing the additional dimensions to zero, with the model's predictions determining the voxel's label. Although this extension may introduce minor inaccuracies in decision boundaries, the approach provides a meaningful and interpretable view of the ML model's decision-making process. Detailed results are provided in Chapter 5, Section 5.1.

4.5 Pipeline of the Methodology

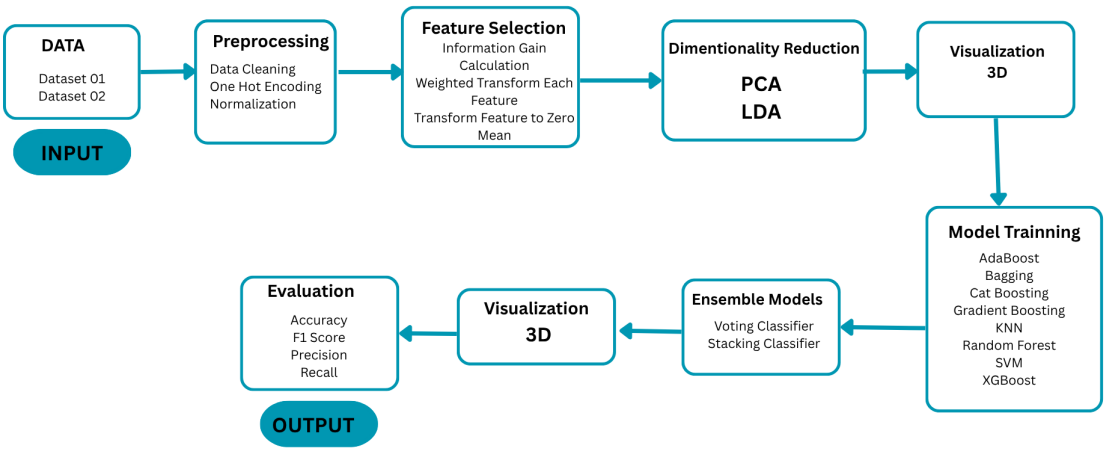


Figure 4.6: Interactive 3D Visualization Framework for Machine Learning-Based Network Intrusion Detection Systems

The 3D Visualization framework presents a structured pipeline for enhancing intrusion detection through optimized feature selection, dimensionality reduction, and interactive visualization. The process involves data preprocessing, feature ranking using information gain, and PCA for reducing dimensionality while preserving critical variance. A Plotly-based 3D visualization enables exploration of class separability and misclassification patterns. Finally, multiple machine learning and ensemble models are trained and evaluated using accuracy, F1-score, precision, and recall, resulting in a robust and interpretable intrusion detection system.

4.6 Algorithm

For real-world attack classification, the pipeline's implementation is described in Algorithm 1. It integrates preprocessing, feature extraction, dimensionality reduction, model training, and evaluation into a unified workflow. Misclassification occurs when the model incorrectly labels a network instance, such as identifying an attack as normal or vice versa. The interactive 3D visualization helps detect these misclassified instances, which typically appear as overlapping regions or deviations between class boundaries. Evaluation metrics, including Accuracy, Precision, Recall and F1-Score, are employed to assess the performance of both base and ensemble models and to determine the system's effectiveness in distinguishing between normal and attack traffic.

Algorithm 1 ML Pipeline: Preprocessing, Feature Extraction, Training, Evaluation, and 3D Visualization

Input: Dataset $D_1 = (X_{\text{train}}, y_{\text{train}})$, $D_2 = (X_{\text{test}}, y_{\text{test}})$
Output: Accuracy, F1 Score, Precision, Recall for all models
Step 1: Preprocessing.
Handle missing values in X_{train}, X_{test}
One-Hot Encode categorical features:
    Apply OHE : $X_{\text{train\_cat}} \leftarrow ohe.fit\_transform(X_{\text{train}})$
    Transform test set : $X_{\text{test\_cat}} \leftarrow ohe.transform(X_{\text{test}})$
Concatenate categorical and numeric arrays : $X_{\text{train\_oh}}, X_{\text{test\_oh}}$
Normalize features:
    Apply Normalization : $X_{\text{train\_normalize}} \leftarrow Scaler.fit\_transform(X_{\text{train\_oh}})$
    Transform test set : $X_{\text{test\_normalize}} \leftarrow Scaler.transform(X_{\text{test\_oh}})$
Step 2: Feature Extraction.
Compute information gain calculation: $IG \leftarrow$ mutual_info_classif($X_{\text{train\_normalized}}$, y_{train})
Calculate weights in $[0, 1]$: $Weights \leftarrow \dfrac{IG}{(IG)_{\max}}$
Weighted transform of features:
    Apply weights :
    $X_{\text{train\_weighted}} \leftarrow X_{\text{train\_normalized}} * Weights$
    $X_{\text{test\_weighted}} \leftarrow X_{\text{test\_normalized}} * Weights$
Transform features to Zero-mean :
    $X_{\text{train\_mean}} \leftarrow X_{\text{train\_weighted}} - mean()$
    $X_{\text{train\_centered}} \leftarrow X_{\text{train\_weighted}} - X_{\text{train\_mean}}$
    $X_{\text{test\_centered}} \leftarrow X_{\text{test\_weighted}} - X_{\text{test\_mean}}$
Step 3: Dimensionality Reduction (PCA,LDA).
    Apply PCA : $X_{\text{train\_pca}} \leftarrow pca.fit\_transform(X_{\text{train\_centered}})$
    Transform test set : $X_{\text{test\_pca}} \leftarrow pca.transform(X_{\text{test\_centered}})$
    Apply LDA : $X_{\text{train\_lda}} \leftarrow lda.transform(X_{\text{train\_pca}})$
                $X_{\text{test\_lda}} \leftarrow lda.transform(X_{\text{test\_pca}})$
Step 4: 3D Visualization.
Step 5: Base ML Model Training & Evaluation.
for each $m \in \{SVM, RF, KNN, GB, XGBoost, AdaBoost, Bagging\}$ **do**
    Train m on $(X_{\text{train\_cfg}}, Y_{\text{train}})$(e.g., $X_{\text{train\_pca}}$ or $X_{\text{train\_lda}}$)
    Predict $\hat{Y}_m \leftarrow m.predict(X_{\text{test\_cfg}})$
    Compute Accuracy, F1 Score, Precision, Recall
end for
Step 6: Ensemble Model Construction & Evaluation.
Build Voting and Stacking Classifiers using trained base models.
for each $e \in \{Voting(hard, soft, weighted), Stacking\}$ **do**
    Train e on $(X_{\text{train\_mixed}}, y_{\text{train}})$
    Predict $\hat{Y}_e \leftarrow e.predict(X_{\text{test\_mixed}})$
    Compute Accuracy, F1, Precision, Recall
end for
3D visualization on model data (13D)
return All evaluation metrics for base and ensemble models.

4.7 ML Model Integration

In the proposed framework, Support Vector Machine (SVM) is used as the main machine learning model to classify network traffic into various categories. The decision space boundaries are created based on the trained SVM model, allowing users to explore the decision-making process visually. The decision space is rendered in 3D using a voxel-based method, enabling an interactive visualization where users can examine the regions where specific categories are correctly classified and identify areas where misclassifications occur. This interactive decision space helps better understand the performance of the machine learning models, providing insights into misclassification patterns that can be used to refine the models.

In addition to SVM, several other machine learning models are integrated into the framework to offer alternative perspectives on classification and evaluate the robustness of the system. These models include:

AdaBoost (Adaptive Boosting):
An ensemble boosting method that combines weak learners sequentially to form a strong classifier. It reduces bias and variance while emphasizing misclassified samples for better accuracy.

Bagging (Bootstrap Aggregating):
Trains multiple models on random data subsets using bootstrapping and aggregates their outputs. It reduces variance and overfitting, improving model stability and generalization.

CatBoost (Categorical Boosting):
A gradient boosting algorithm optimized for categorical data with minimal preprocessing. It is efficient, avoids overfitting, and performs well on mixed-type datasets.

Gradient Boosting (GB):
Builds models sequentially by minimizing residual errors via gradient descent. It provides high accuracy and handles complex nonlinear relationships effectively.

K-Nearest Neighbors (KNN):
A simple instance-based algorithm that classifies samples based on the majority of their k nearest neighbors. It is easy to implement and adapts well to irregular decision boundaries.

Random Forest (RF):
An ensemble of multiple decision trees trained on random subsets of data and features. It reduces bias and variance, handles imbalance, and provides feature importance.

Support Vector Machine (SVM):
Finds an optimal hyperplane to separate classes in high-dimensional space using kernel functions. Performs well with small, high-dimensional, and non-linear datasets.

XGBoost (Extreme Gradient Boosting):
An optimized gradient boosting framework using regularization and second-order derivatives. It is fast, scalable, and prevents overfitting for superior performance.

Ensemble Models:
Combine multiple base learners (e.g., DT, SVM, KNN) to improve robustness and accuracy. They reduce bias, variance, and enhance model generalization.

Voting Classifier:
Aggregates predictions of diverse models using majority (hard) or probability (soft) voting. Improves performance through complementary model strengths.

Stacking Classifier:
Combines several base models and a meta-model that learns to merge their outputs. Achieves higher accuracy and generalization by leveraging heterogeneous learners.

By integrating these models into the framework, the system offers flexibility in model selection and comparison. The goal is to take advantage of the strengths of each model to improve the accuracy of the classification and analyze misclassifications more thoroughly.

CHAPTER V

RESULTS AND DISCUSSION

This chapter presents the results and analysis of the proposed framework using both visual and quantitative evaluations to assess its overall performance and interpretability. It includes an interactive 3D visualization of network intrusion detection data from both datasets and trained models, illustrating how the system represents data distributions, decision boundaries, and classification behavior in three-dimensional space. The visualization also helps identify non-homogeneous clusters and instances of misclassification across eight base models and two ensemble models. In addition to visual analysis, a statistical evaluation is conducted using standard metrics such as Accuracy, Precision, Recall, and F1-Score to compare the performance and reliability of the models. The findings are then discussed in relation to the research objectives, highlighting the causes of misclassification, the strengths and limitations of the proposed framework, and potential directions for enhancing the interpretability and effectiveness of machine-learning-based intrusion detection systems.

5.1 Visualization Results

we present a 3D visualization framework designed to evaluate the performance of 8 base models and 2 ensemble models on the UNSW-NB15 and NSL-KDD datasets. The system provides real-time, interactive exploration, allowing users to navigate the 3D space with six degrees of freedom, enabling them to closely examine decision boundaries and model classifications.

By visualizing these decision boundaries, the framework facilitates a comparative analysis of the individual base models' classification performance. It also demonstrates how the ensemble models, by integrating the outputs of the base models, enhance classification accuracy. This interactive exploration helps identify areas of misclassification, offering valuable insights into the model's behavior and highlighting the effectiveness of ensemble methods in improving network intrusion detection.

5.1.1 UNSW-NB15

Figure 5.1 presents a 3D visualization of the UNSW-NB15 [15] dataset, with separate views for the training and testing sets.

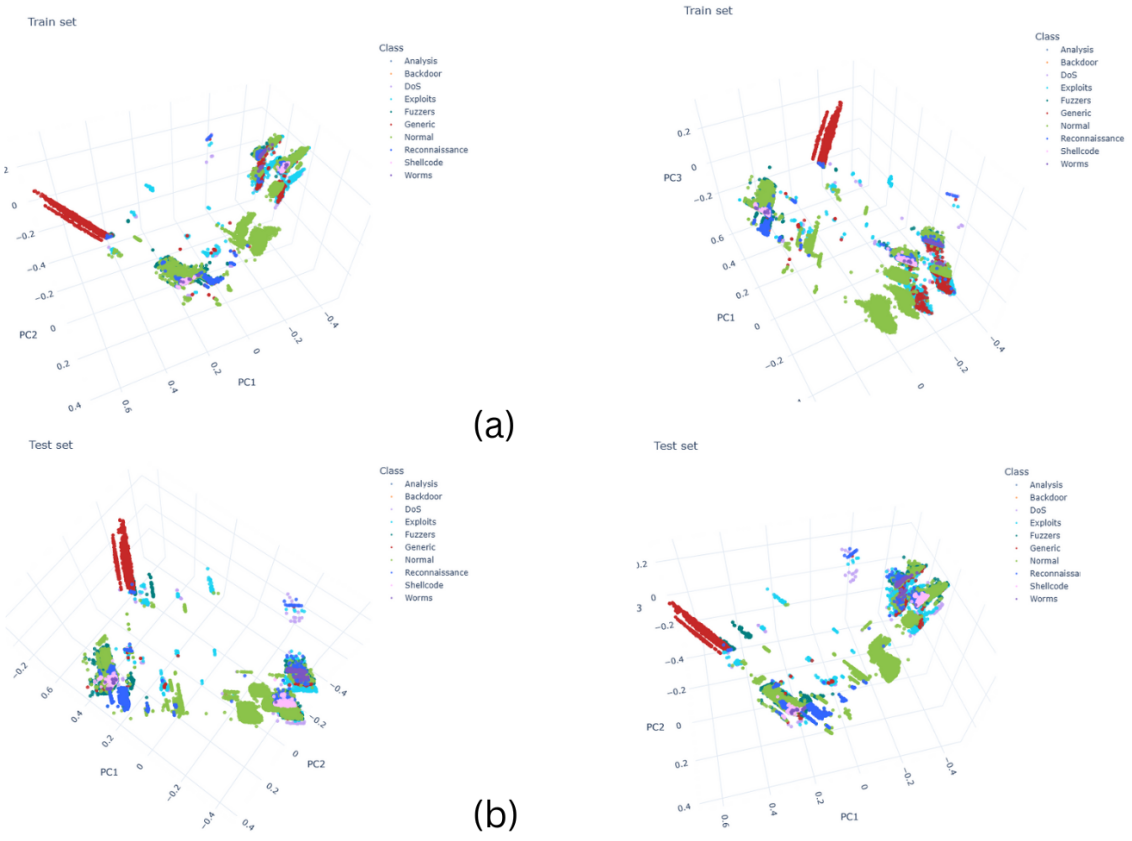


Figure 5.1: 3D visual representation of the UNSW-NB15 dataset; (a) data from the training set from two different camera viewpoints; (b) data from the testing set from two different camera viewpoints.

Fig. 5.1(a) shows a portion of the training set from two viewpoints, with network traffic categories color-coded. Traffic from the same category clusters together in the 3D space.

Fig. 5.1(b) displays a similar 3D visualization of the testing set from two distinct viewpoints. Comparing the training and testing sets reveals significant similarity in their patterns, suggesting that a machine learning model trained on the training set is likely to generalize well to the testing set. The consistency between the two sets indicates the model will effectively identify attack instances in the testing data.

Users can interactively select specific portions of the data to retrieve statistical information and extract the selected data from the visual representation into a text format for further analysis. Fig. 5.2(a) illustrates how users can define a region by selecting a group of data points within an adjustable radius around a reference point to obtain relevant information about the group. In contrast, Fig. 5.2(b) presents a scenario where a single data point is selected.

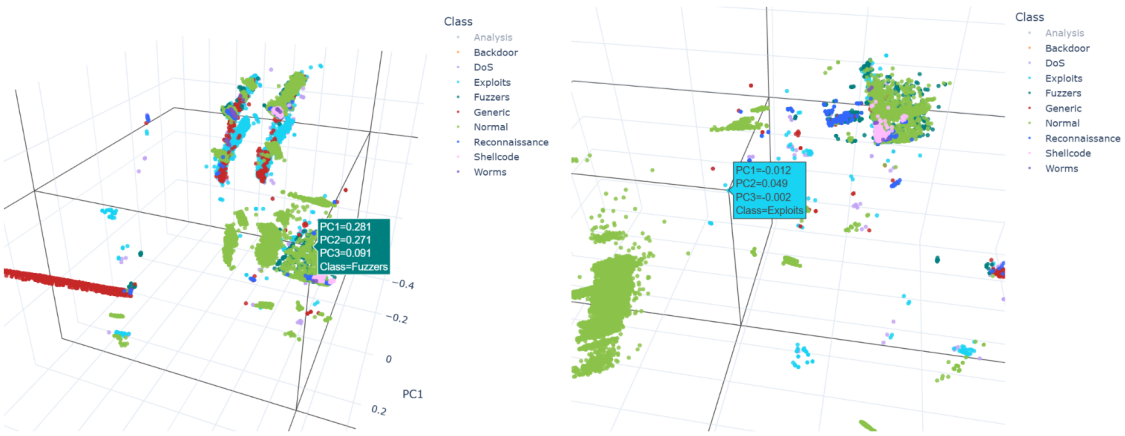


Figure 5.2: Information on selected data; (a) selecting a group of data points; (b) selecting a single data point.

Once the data is selected, users can extract detailed information pertaining to the chosen data points for closer inspection.

The data groupings reveal that traffic associated with generic attacks is well-clustered, while mixed clusters contain both attack and normal traffic, making them less distinct. Fig. 5.3 shows examples of these clusters, with Figs. 5.3(a) and 5.3(b) illustrating normal traffic clusters from the training and testing sets, respectively, demonstrating a high similarity between the two.

For machine learning (ML), traffic within homogeneous clusters is easier to identify, leading to better detection performance. Figs. 5.3(c) and 5.3(d) show clusters containing generic attacks, which are mostly homogeneous and thus easier for ML models to detect. This is supported by previous studies.

Conversely, mixed clusters, shown in Figs. 5.3(e) and 5.3(f), contain both normal and abnormal traffic, making accurate classification difficult for ML models. Additionally, the training and testing sets differ in the number of worm attacks, and the presence of many exploit attacks complicates classification, increasing the risk of misclassification.

Given the similar patterns between the training and testing sets (Figs. 5.1(a) and 5.1(b)), it is expected that a feature transformation approach that organizes the training data into homogeneous clusters will similarly group the testing data.

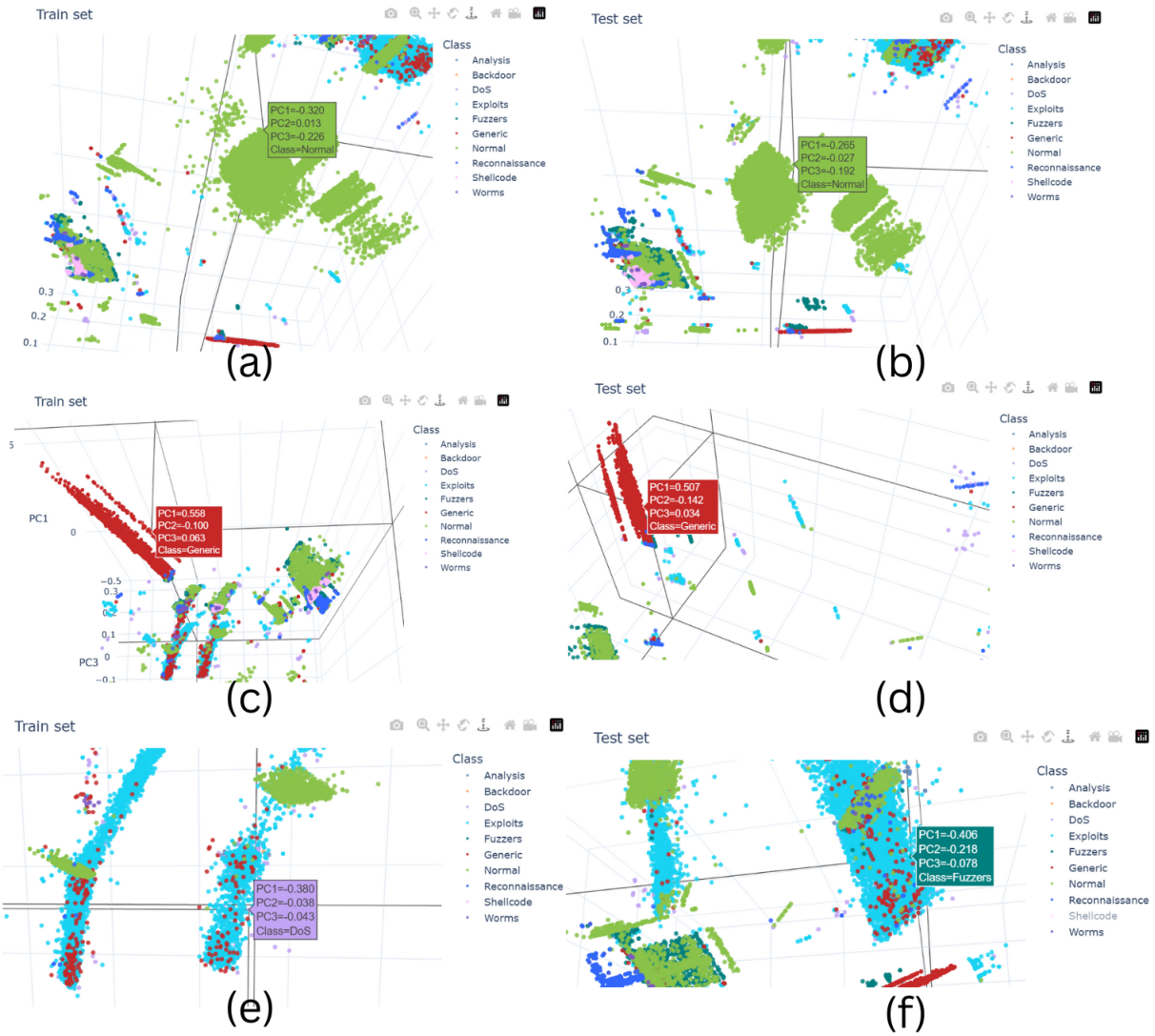


Figure 5.3: Example clusters from the UNSW-NB15 dataset; (a)–(b) a cluster containing normal traffic from the training and testing data, respectively; (c)–(d) a cluster containing mostly generic attacks from the training and testing data, respectively; (e)–(f) a cluster containing mixed traffic from the training and testing data, respectively.

Experiments were conducted to visualize the 3D machine learning (ML) decision space using the approach outlined in Chapter 4, Section 4.4,Section 4.4. A Support Vector Machine (SVM) with C-support vector classification was trained on 13-dimensional (13D) training data and used to detect attacks in the testing set. The SVM was implemented using default parameters (kernel = "rbf" , gamma = "scale", C = 1).

To demonstrate the results, binary classification was first applied, where network traffic was classified as either normal or abnormal. The confusion matrix for binary classification is shown in Table 2, and the decision space visualization for the trained SVM is depicted in Fig. 5.4. Fig. 5.4(a) provides an overview, while Figs. 5.4(b) and 5.4(c) show the decision spaces for normal and abnormal traffic, respectively.

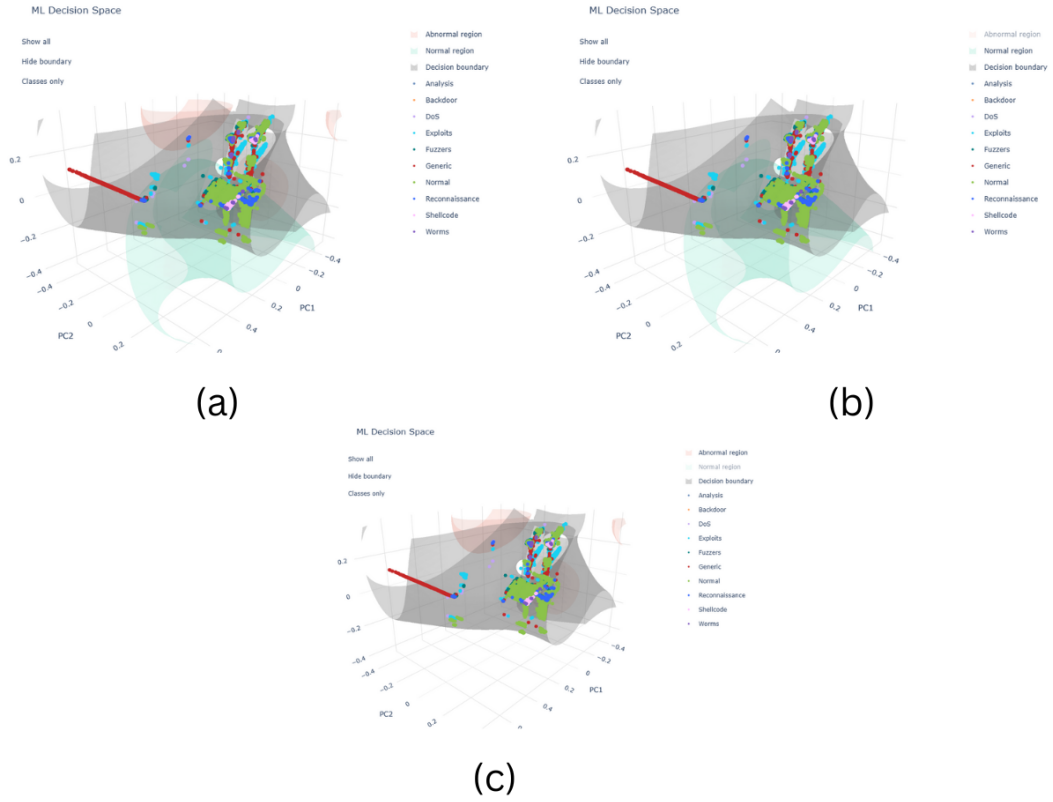


Figure 5.4: Binary classification decision space for the UNSW-NB15 dataset; (a) overview of the decision space; (b) decision space for normal traffic; (c) decision space for abnormal traffic.

Table 5.1: Binary classification confusion matrix for the UNSW-NB15 dataset.

Predicted		**Normal**	**Abnormal**	**Recall (%)**
	Normal	56,000	0	100.0
Actual	**Abnormal**	1,010	118,331	99.2
	Precision (%)	98.2	100.0	

Instances within a decision space are classified according to the type of that space. For example, instances in the normal traffic space are classified as normal. Table 5.1 shows that although the recall rate for abnormal traffic is high, nearly half of the normal traffic was misclassified as abnormal. This misclassification is due to mixed clusters that the model struggles to distinguish. The interactive visualization approach facilitates inspection of this issue.

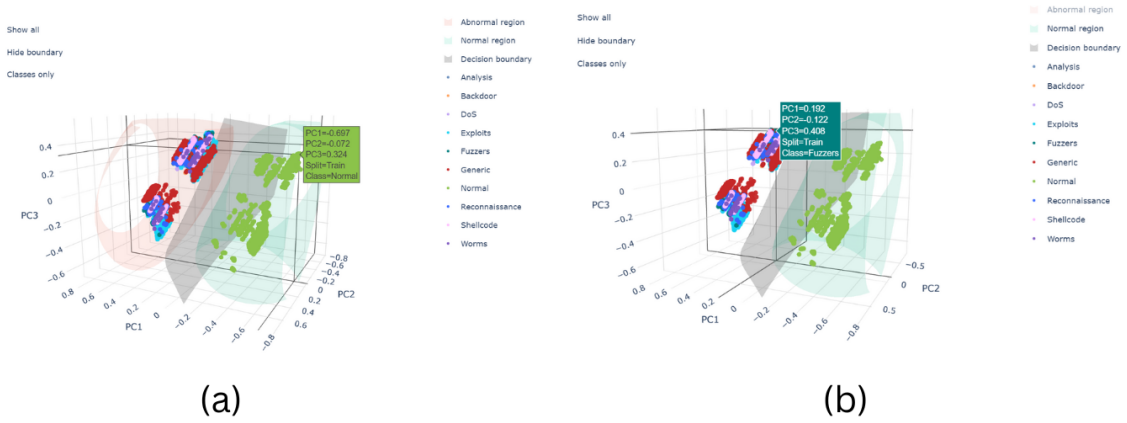


Figure 5.5: Close-up showing the decision space boundary and instances of misclassified traffic; (a) mixture of normal and abnormal traffic outside the decision space; (b) normal traffic misclassified as abnormal traffic.

Fig. 5.5 presents a close-up visualization of the misclassification. In Fig. 5.5(a), the normal traffic decision space is visible on the right, correctly classifying the normal traffic within that space. However, a cluster on the left contains both normal and abnormal traffic, lying outside the normal space. Fig. 5.5(b) highlights the normal traffic instances misclassified as abnormal, showing that many normal instances were incorrectly classified as abnormal.

Experiments were conducted to visualize the decision spaces for multi-category classification, where traffic is classified into categories such as normal, DoS, worm, exploits, etc., as opposed to binary classification, which only classifies traffic as normal or abnormal. Figure 5.6 illustrates the decision spaces for various categories, with Fig. 5.6(a) providing an overview, and Figs. 5.6(b) to 5.6(g) showing decision spaces for specific categories: normal traffic, fuzzers, exploits, generic attacks, reconnaissance, and DoS traffic.

The 3D visualization reveals that most generic attack instances fall within the generic attack decision space. The decision spaces for reconnaissance and DoS attacks (Figs. 5.6(f) and 5.6(g)) are relatively small, with close-up views provided. Some categories, such as analysis and backdoor traffic, were not visualized due to the basic SVM model's inability to detect these attacks. Moreover, the higher proportion of exploit attacks in certain clusters led to the misclassification of other traffic as exploits. This is evident in Fig. 5.7, where Fig. 5.7(a) shows close-up views of the exploits decision space, highlighting the mixture of exploits and other traffic, and Fig. 5.7(b) shows the misclassified traffic instances from two viewpoints.

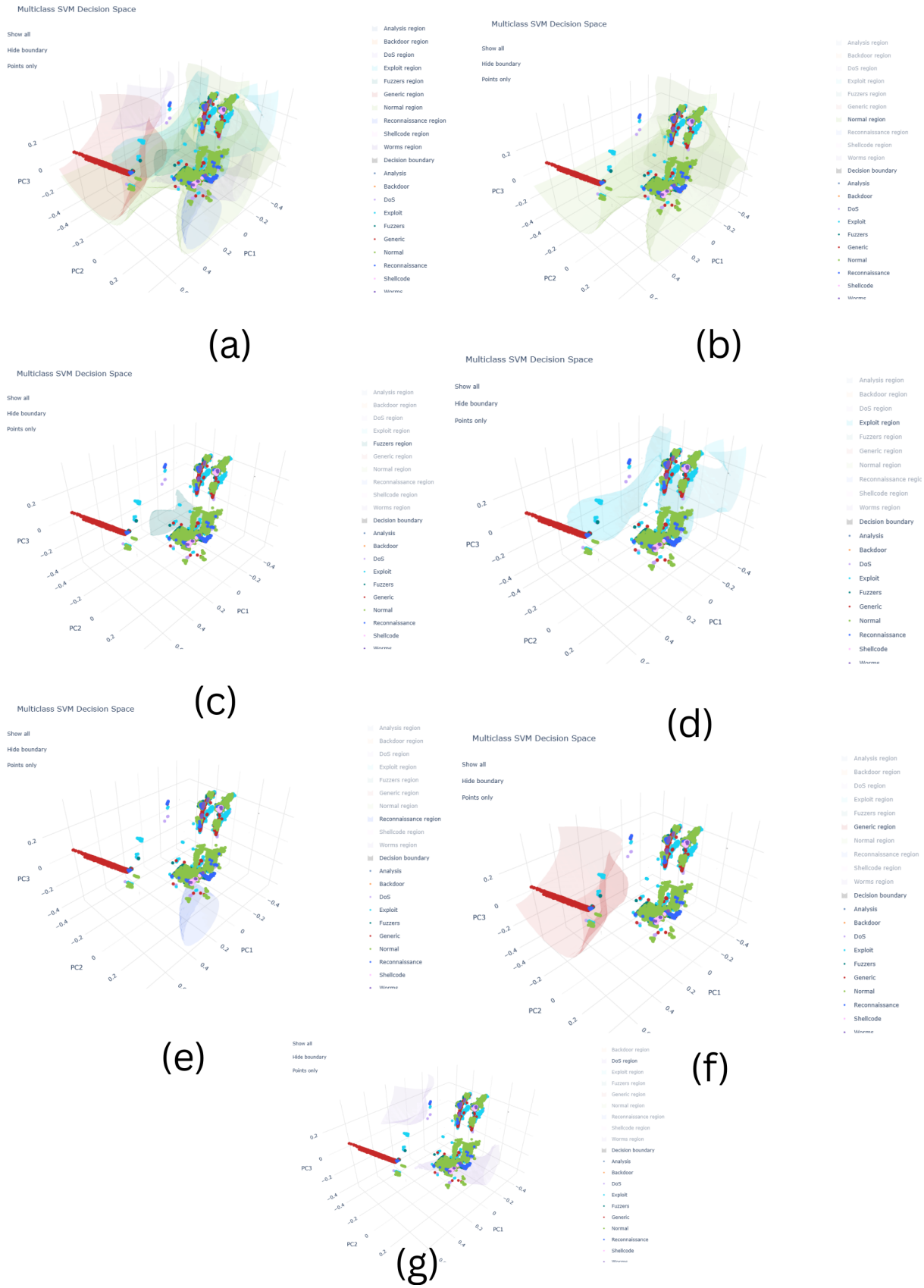
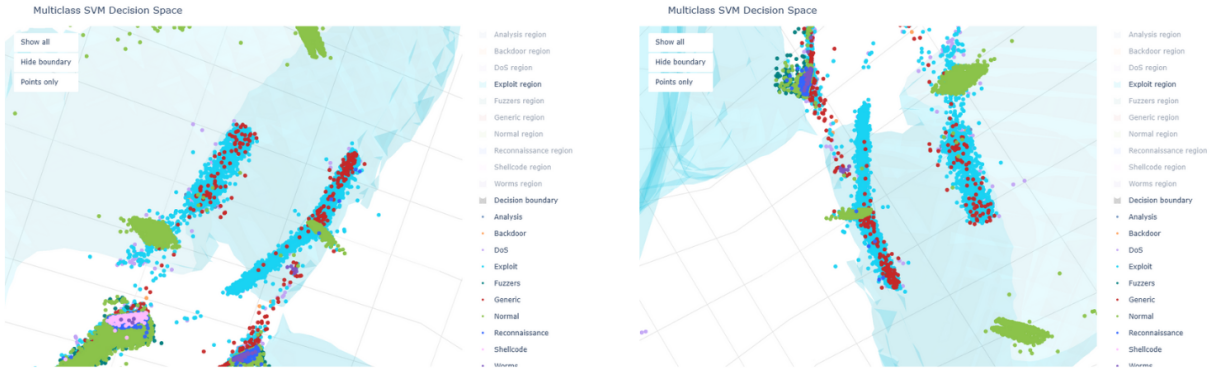


Figure 5.6: Multi-category classification decision spaces for the UNSW-NB15 dataset; (a) overview of the decision spaces; (b) decision space for normal traffic; (c) decision space for fuzzers; (d) decision space for exploits; (e) decision space for generic attacks; (f) decision space for reconnaissance traffic; (g) decision space for DoS traffic.



(a) (b)

Figure 5.7: Close-up showing a part of the decision space for exploits and instances of misclassified traffic; (a) mixture of exploits and other traffic inside the exploits decision space from two different camera viewpoints; (b) traffic that was misclassified as exploits from two different camera viewpoints.

Experiments were conducted to visualize the decision spaces of both base machine learning (ML) models and ensemble models using the UNSW-NB15 dataset. Fig. 5.8 illustrates the decision spaces for the base models: Fig. 5.8(a) for Random Forest, Fig. 5.8(b) for K-Nearest Neighbors (KNN), Fig. 5.8(c) for Gradient Boosting, Fig. 5.8(d) for XGBoost, Fig. 5.8(e) for AdaBoost, Fig. 5.8(f) for CatBoost, and Fig. 5.8(g) for Bagging Classifier. These visualizations show how each model handles misclassification in non-homogeneous clusters, where overlapping categories create ambiguity.

Fig. 5.9 depicts the decision spaces for the ensemble models: Fig. 5.9(a) for the Voting Classifier and Fig. 5.9(b) for the Stacking Classifier. Ensemble methods, which combine multiple base models, generate more generalized decision spaces, improving misclassification handling and reducing overfitting compared to individual base models.

These 3D visualizations allow for a direct comparison between the decision boundaries of base models and ensemble models on the UNSW-NB15 dataset. Overlapping decision spaces highlight areas of misclassification, particularly in non-homogeneous clusters where the models struggle to distinguish between categories. The results demonstrate the advantages of ensemble methods in managing complex decision spaces and improving classification accuracy in multi-class tasks.

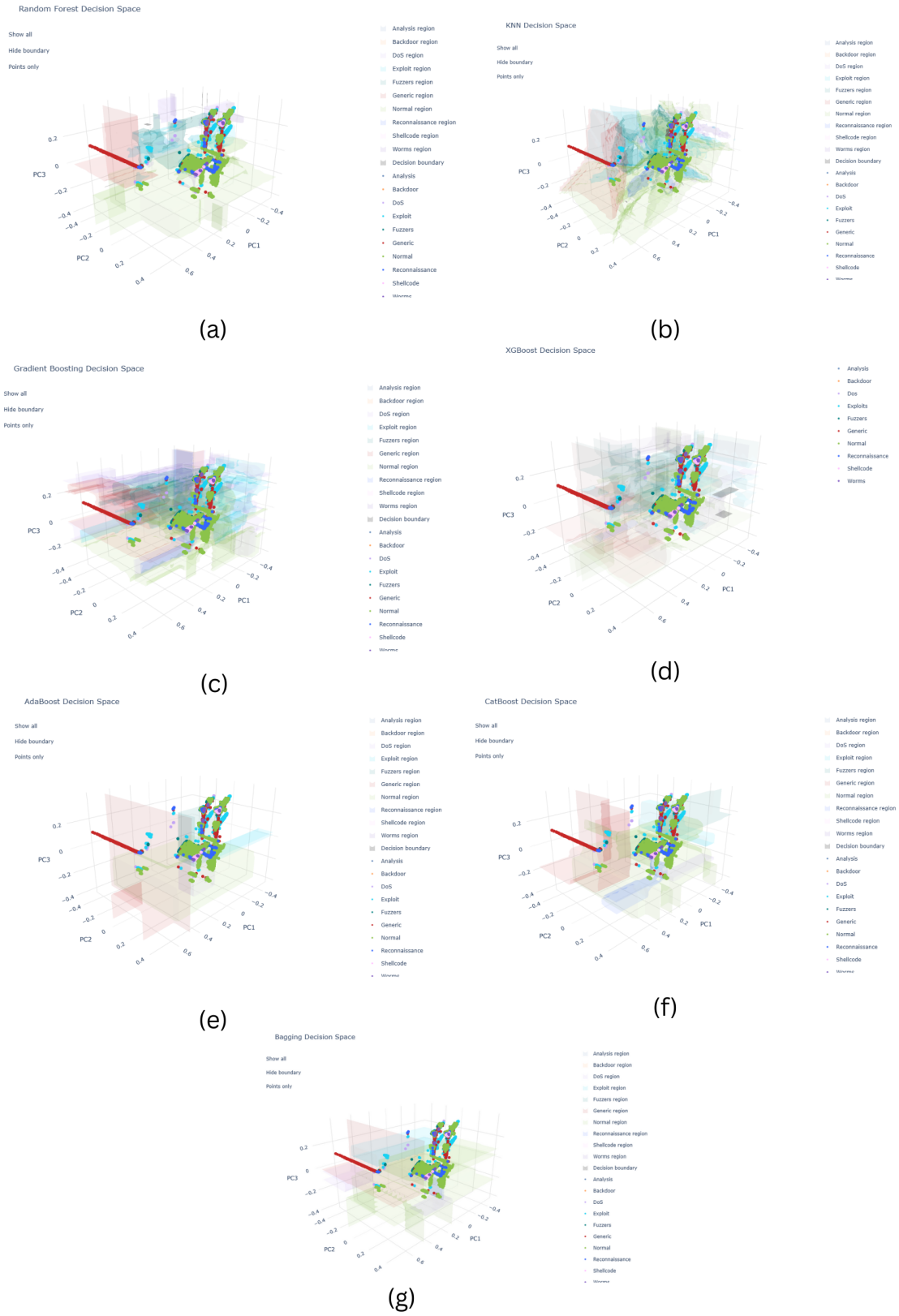


Figure 5.8: Base ML model decision spaces for the UNSW_NB15 dataset; (a) Decision spaces for the Random Forest; (b) decision space for KNN; (c) decision space for Gradient Boosting; (d) decision space for XGBoosting. (e) decision space for AdaBoosting; (f) decision space for CatBoosting; (g) decision space for Bagging.

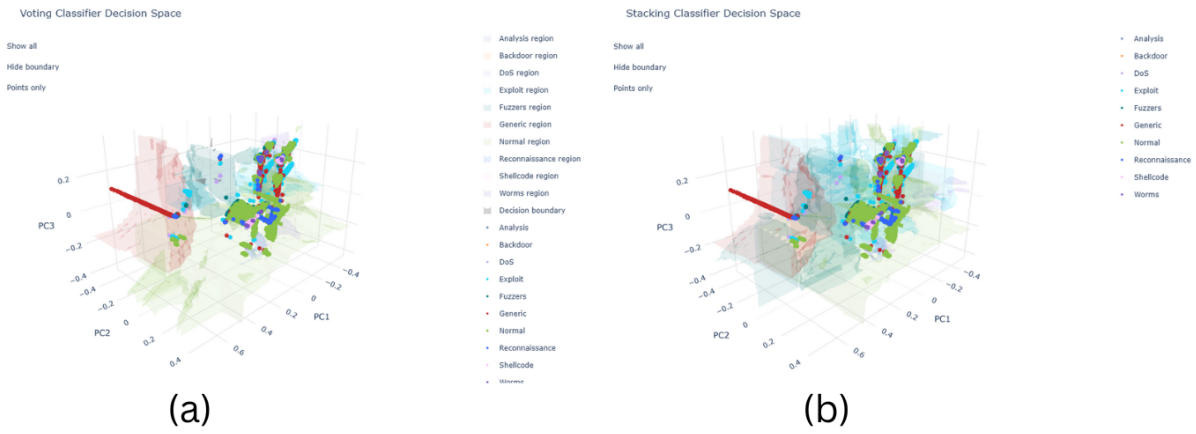


Figure 5.9: Ensamble model decision spaces for the UNSW_NB15 dataset; (a) decision space for voting classifier; (b) decision space for Stacking classifier.

5.1.2 NSL-KDD

Fig. 5.10 provides a 3D visualization of the NSL-KDD [14] dataset, as rendered by the proposed system. Fig. 5.10(a) illustrates a subset of the training set, while Fig. 5.10(b) shows a portion of the testing set, with both visualizations captured from two distinct camera viewpoints.

While the overall visual distributions of the training and testing sets exhibit some similarities, notable differences emerge in the NSL-KDD dataset compared to the UNSW-NB15 dataset. Specifically, the traffic characteristics in the training and testing sets of the NSL-KDD dataset demonstrate more pronounced divergence, whereas the UNSW-NB15 dataset exhibits greater similarity between the training and testing sets in terms of traffic patterns.

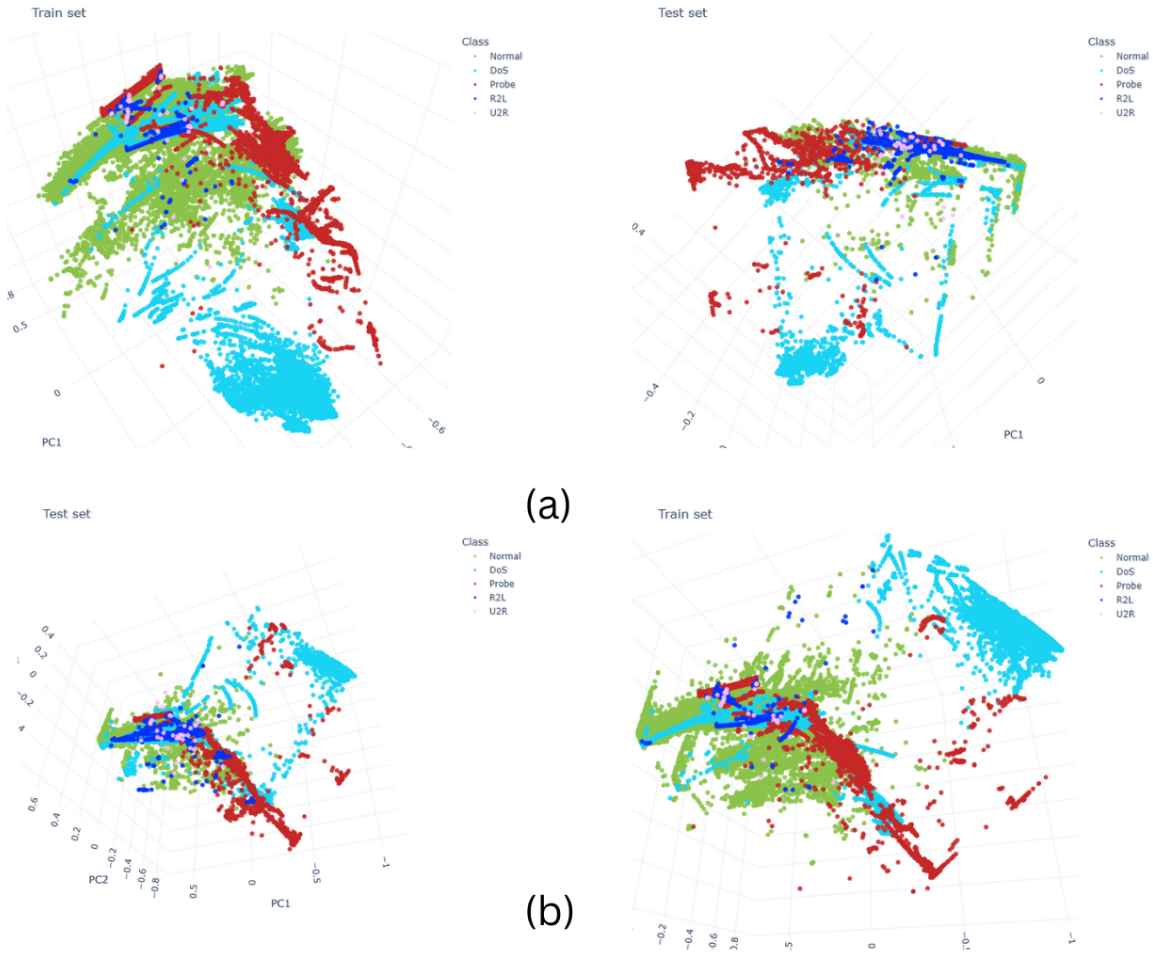


Figure 5.10: 3D visual representation of the NSL_KDD dataset; (a) data from the training set from two different camera viewpoints; (b) data from the testing set from two different camera viewpoints.

Users are provided with the ability to interactively select specific regions of the data to retrieve statistical information and export the selected data from the visual representation into a text format for further analysis. Fig. 5.11(a) demonstrates how users can define a region by selecting a group of data points within an adjustable radius around a reference point, allowing for the extraction of relevant details regarding the selected group. In contrast, Fig. 5.11(b) depicts a scenario in which a single data point is selected. Once a data point is chosen, users can extract detailed information associated with the selected point for closer inspection.

The visual representation reveals clusters of both homogeneous records and diverse traffic types. Figs. 5.12(a) and 5.12(b) display DoS attack traffic from the training and testing sets, respectively. The DoS attack cluster is clearly isolated and primarily composed of homogeneous records, facilitating accurate identification by machine learning (ML) models.

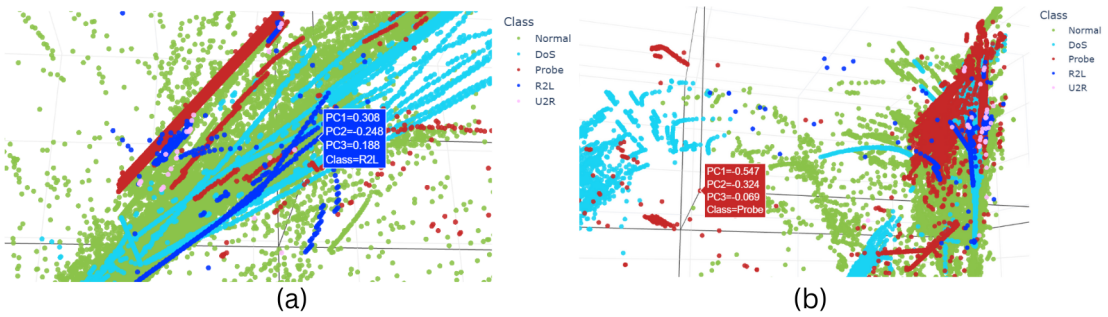


Figure 5.11: Information on selected data; (a) selecting a group of data points; (b) selecting a single data point.

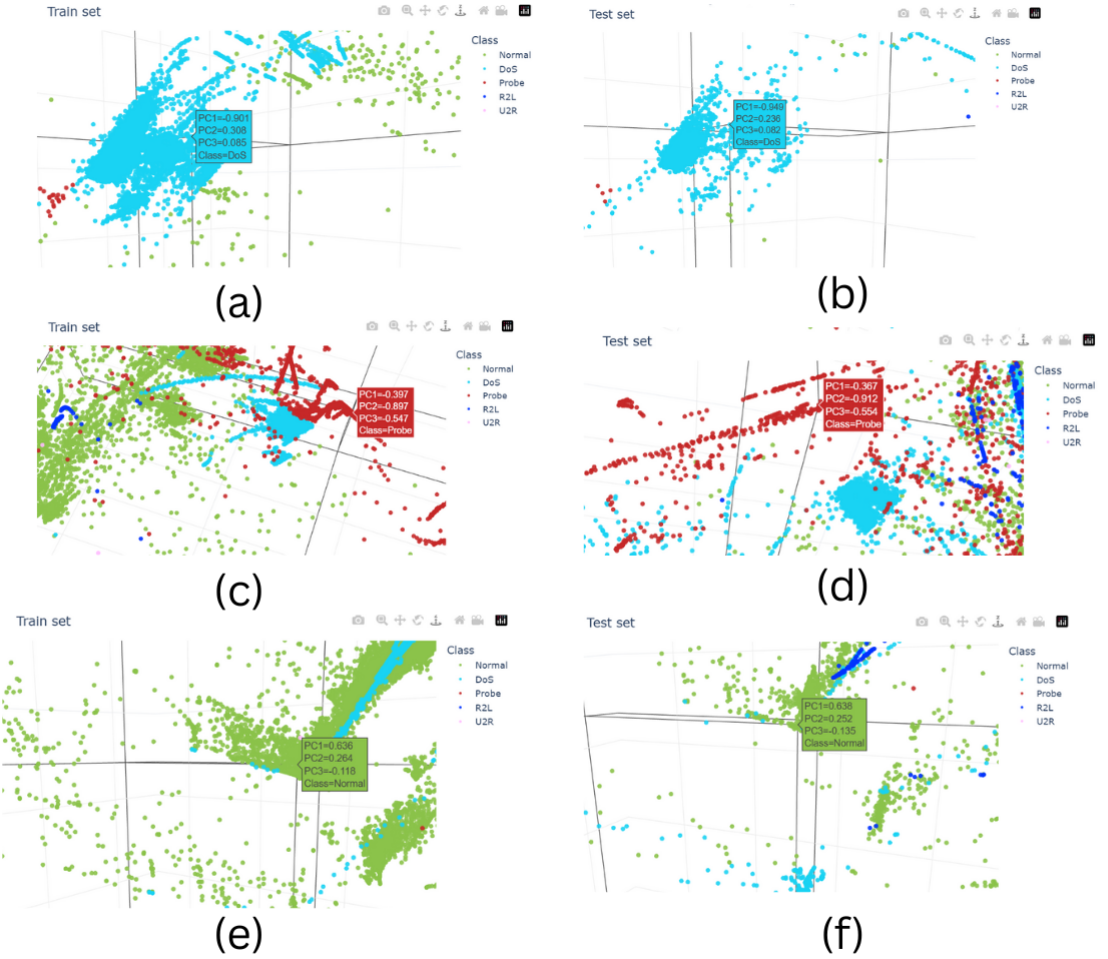


Figure 5.12: Examples of clusters from the NSL_KDD dataset; (a)–(b) a cluster containing DoS attacks from the training and testing data, respectively; (c)–(d) a cluster containing mostly probe attacks from the training and testing data, respectively; (e)–(f) a cluster containing mainly normal traffic from the training and testing data, respectively.

Similarly, Figs. 5.12(c) and 5.12(d) show probe attack traffic in the training and testing sets. While Fig. 5.12(d) depicts some overlap with normal traffic, the majority of traffic within the cluster consists of probe attacks. However, ML models may misclassify adjacent traffic, such as normal traffic or other attacks, as probe attacks in the testing set.

A significant challenge for ML techniques in the NSL-KDD dataset arises from the presence of previously unknown attacks in the testing set. Although the attack categories remain consistent between the training and testing sets, the traffic characteristics exhibit notable differences. This is illustrated in Figs. 5.12(e) and 5.12(f), where Fig. 5.12(e) shows a minimal presence of R2L attacks in the training set, while Fig. 5.12(f) highlights a significant number of R2L attacks in the corresponding testing set. ML models that fail to account for these discrepancies in attack characteristics may struggle to detect these variations, resulting in suboptimal performance.

The same set of experiments conducted on the UNSW-NB15 dataset was also performed on the NSL-KDD dataset to visualize the decision space. The decision space visualization for binary classification is presented in Fig. 13. Fig. 13(a) provides an overview of the binary decision space, while Figs. 13(b) and 13(c) depict the decision spaces for normal and abnormal traffic, respectively.

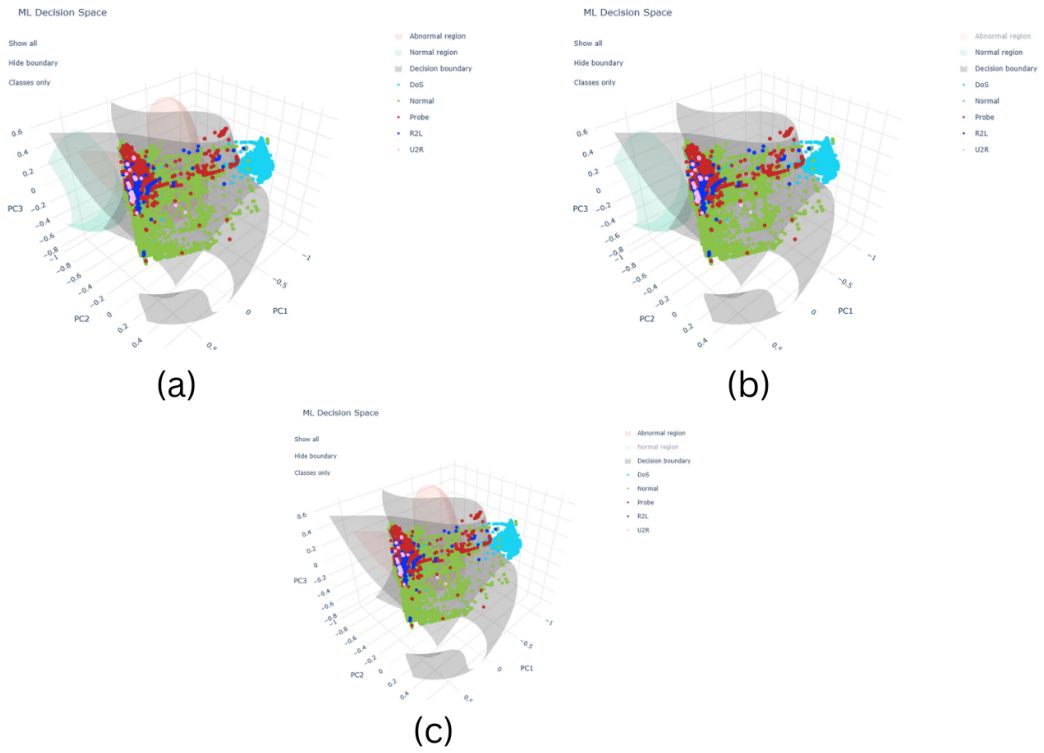


Figure 5.13: Binary classification decision space for the NSL_KDD dataset; (a) overview of the decision space; (b) decision space for normal traffic; (c) decision space for abnormal traffic.

Table 5.2: Multi-category classification confusion matrix for the NSL_KDD dataset.

	Predicted					
Actual	**DoS**	**Normal**	**Probe**	**R2L**	**U2R**	**Recall (%)**
DoS	7443	627	25	0	0	91.9
Normal	173	10972	97	3	0	97.6
Probe	66	266	1825	0	0	84.6
R2L	1	902	49	57	0	5.6
U2R	0	37	1	0	0	0.0
Precision (%)	**96.9**	**85.7**	**91.4**	**95.0**	**0.0**	

The results of multi-category classification are presented in Table 5.2, which includes the confusion matrix, while the decision spaces are illustrated in Fig. 5.14. Fig. 5.14(a) provides an overall depiction of the decision spaces, while Figs. 5.14(b) to 5.14(e) show the individual decision spaces for normal traffic, DoS attacks, R2L attacks, and probe attacks, respectively. Notably, although a distinct U2R decision space is not visible in the 3D visualization, some U2R attacks can still be accurately classified in the 13D space, as demonstrated in Table 5.2.

Experiments were conducted to visualize the decision spaces of both base machine learning (ML) models and ensemble models using the NSL-KDD dataset, following a similar approach as applied to the UNSW-NB15 dataset. Fig. 5.15 illustrates the decision spaces for the base models: Fig. 5.15(a) for Random Forest, Fig. 5.15(b) for K-Nearest Neighbors (KNN), Fig. 5.15(c) for Gradient Boosting, Fig. 5.15(d) for XGBoost, Fig. 5.15(e) for AdaBoost, Fig. 5.15(f) for CatBoost, and Fig. 5.15(g) for Bagging Classifier. These visualizations demonstrate how each model handles misclassification, particularly in non-homogeneous clusters, where overlapping categories create ambiguity.

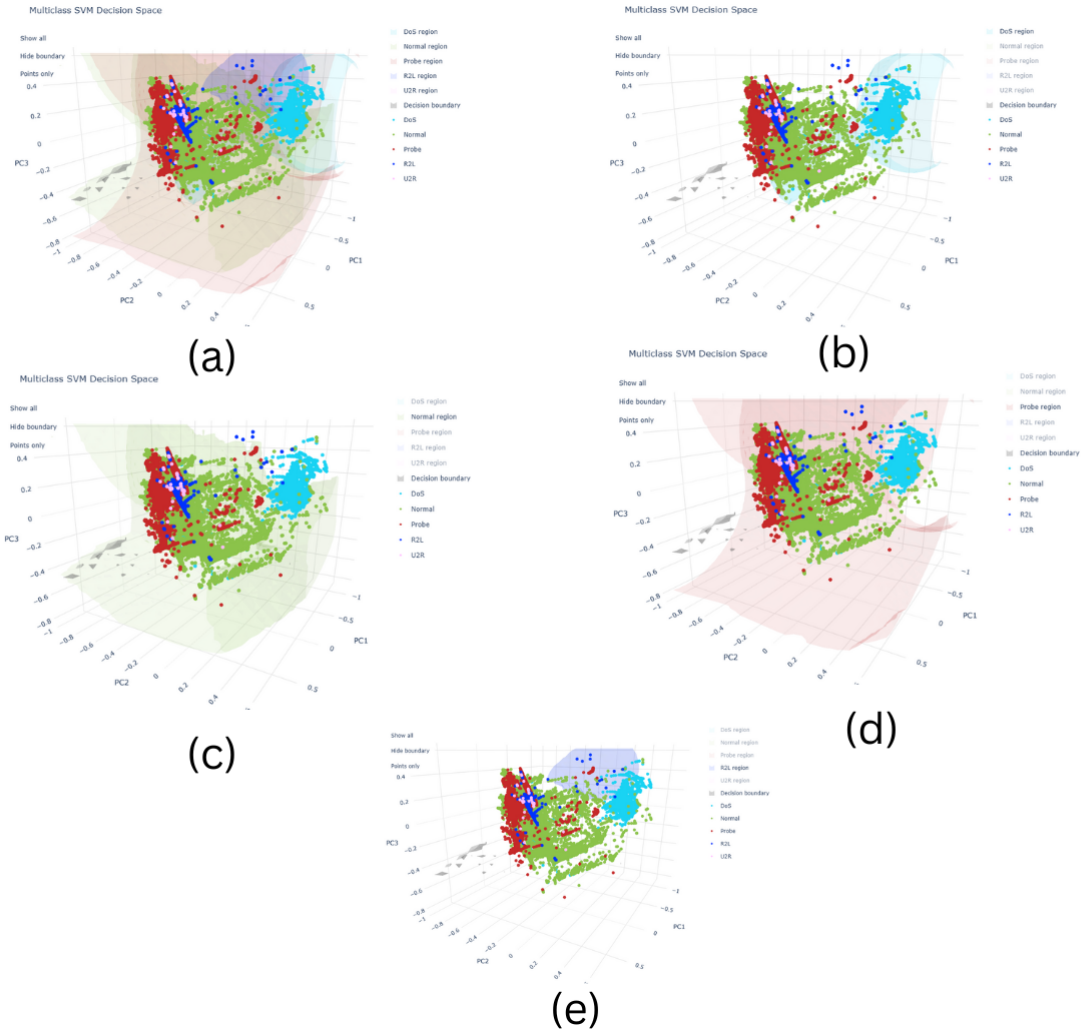


Figure 5.14: Multi-category classification decision spaces for the NSL_KDD dataset; (a) overview of the decision spaces; (b) decision space for DoS traffic; (c) decision space for normal attacks; (d) decision space for probe attacks. (e) decision space for R2L attacks.

Fig. 5.16 presents the decision spaces for the ensemble models: Fig. 5.16(a) for the Voting Classifier and Fig. 5.16(b) for the Stacking Classifier. Ensemble methods, which combine multiple base models, produce more generalized decision spaces, thereby improving the handling of misclassifications and reducing overfitting compared to individual base models.

These 3D visualizations enable a direct comparison between the decision boundaries formed by base and ensemble models on the NSL-KDD dataset. Areas of overlap between decision spaces indicate regions of misclassification, particularly in non-homogeneous clusters, where models struggle to distinguish between categories. The results highlight the advantages of ensemble methods in managing complex decision spaces and improving classification accuracy in multi-class tasks.

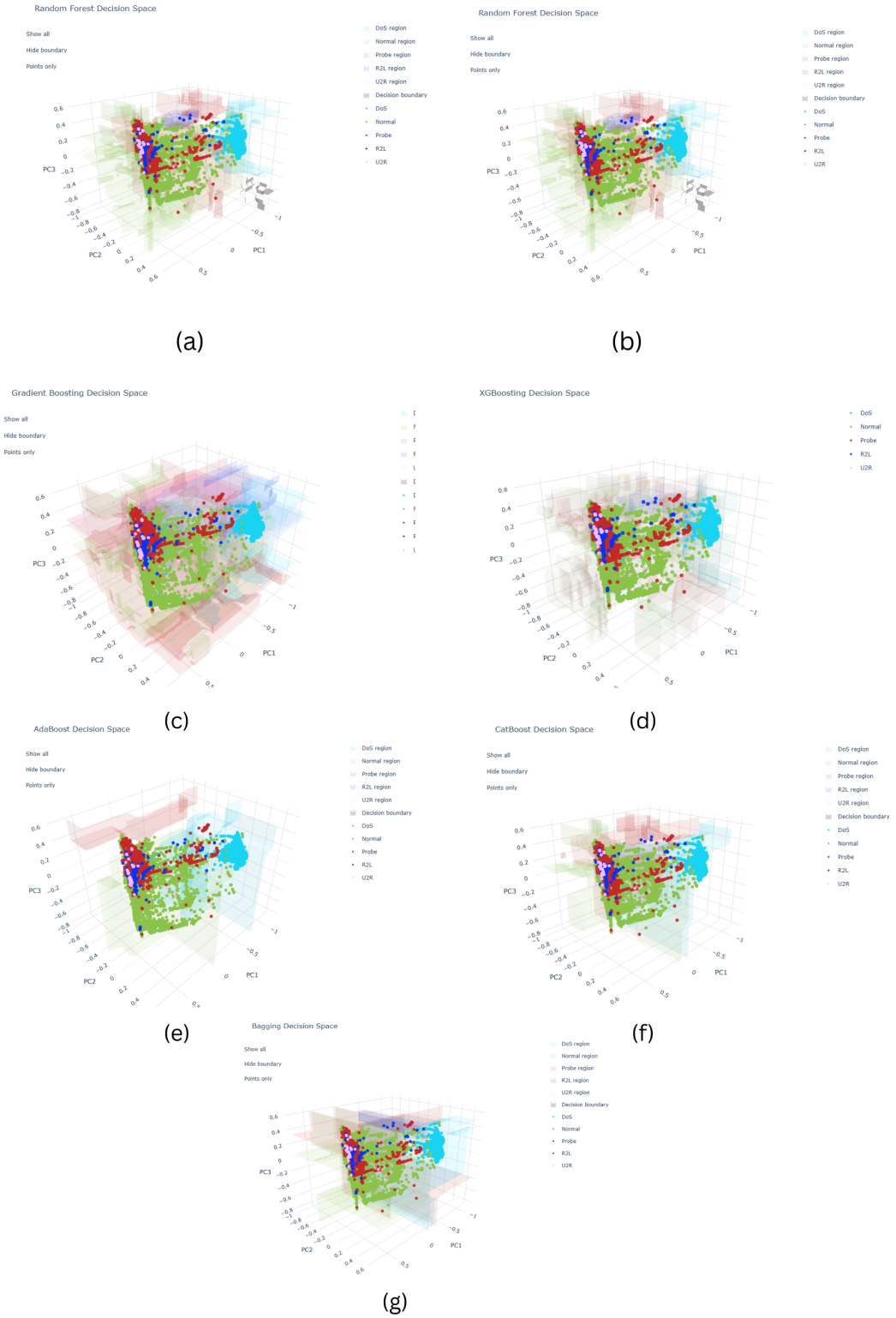


Figure 5.15: Base ML model decision spaces for the NSL_KDD dataset; (a) Decision spaces for the Random Forest; (b) decision space for KNN; (c) decision space for Gradient Boosting; (d) decision space for XGBoosting. (e) decision space for AdaBoosting; (f) decision space for CatBoosting; (g) decision space for Bagging.

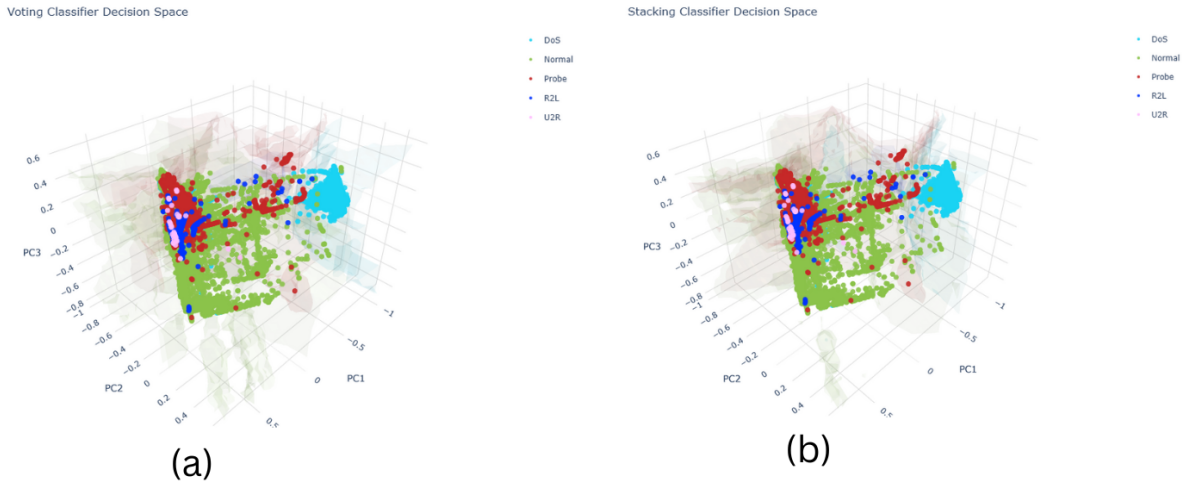


Figure 5.16: Ensamble model decision spaces for the NSL_KDD dataset; (a) decision space for voting classifier; (b) decision space for Stacking classifier.

5.1.3 NSL-KDD Merge

Fig. 17(a) presents a 3D visualization of the training set, which constitutes 75% of the dataset resulting from the merging of the existing NSL-KDD training and test sets. After merging, the data is split into 75% for training and 25% for testing. The points are then projected onto the first three principal components (PC1, PC2, and PC3) following Principal Component Analysis (PCA) for dimensionality reduction. Fig. 17(b) illustrates the 3D representation of the test set, corresponding to the remaining 25% of the merged dataset. Both visualizations highlight the distinct separation between different classes, with each data point color-coded by its class label. However, despite the apparent separation, the potential for misclassification remains, particularly due to the complex nature of the dataset. Additionally, the reduced feature space sacrifices interpretability, as the principal components abstract the original features, making it difficult to directly interpret the relationships between features and class labels. These challenges underscore the limitations of applying PCA for dimensionality reduction in terms of both classification accuracy and model interpretability.

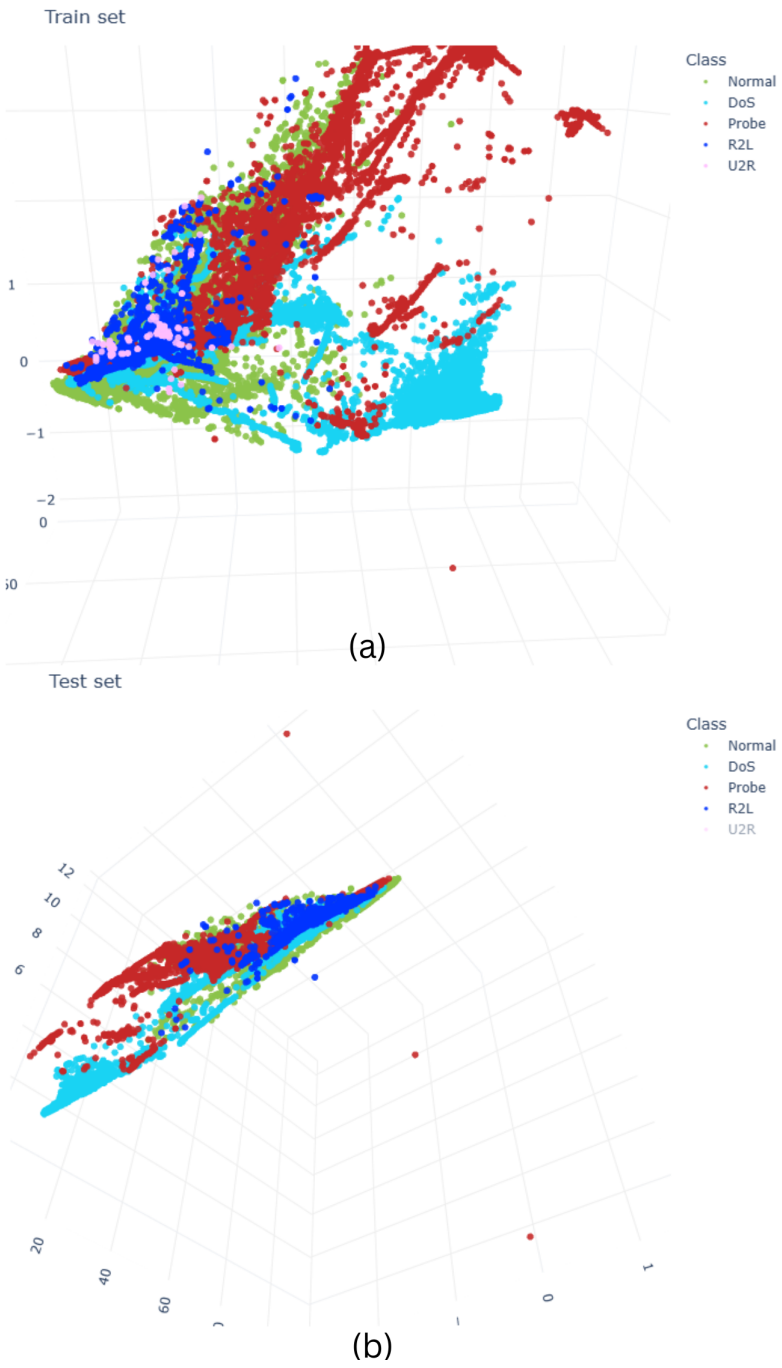


Figure 5.17: 3D visual representation of the NSL-KDD Merge dataset; (a) data from the training set; (b) data from the testing set.

5.2 Statistical Evaluation and Performance Analysis

Table 5.3: Evaluation Metrics Across Two Datasets (UNSW-NB15 and NSL-KDD)

ML Model	Configuration	UNSW-NB15				NSL-KDD			
		Accuracy	F1 Score	Recall	Precision	Accuracy	F1 Score	Recall	Precision
AdaBoosting	PCA, LDA	99.088	98.959	99.330	98.612	86.063	51.946	51.510	62.861
Bagging Classifier	PCA, LDA	99.088	98.959	99.338	98.612	88.303	55.085	54.415	72.232
CatBoosting	PCA, LDA	99.987	99.986	99.980	99.991	88.267	55.237	54.504	74.071
Gradient Boosting	PCA	99.088	98.959	99.330	98.612	91.812	64.716	60.998	91.538
K Nearest Neighbors (KNN)	PCA, LDA	**100.000**	100.000	100.000	100.000	91.443	69.935	64.722	95.183
Random Forest	PCA	99.252	99.145	99.451	98.856	91.789	70.735	65.054	95.808
Support Vector Machine (SVM)	PCA	99.424	99.341	99.577	99.114	90.033	56.825	55.955	73.791
XGBoosting	PCA, LDA	99.968	99.963	99.977	99.950	**92.184**	70.105	64.695	94.201

At a Glance :
(i) For the UNSW-NB15 dataset, the K-Nearest Neighbors (KNN) model achieved the highest accuracy of 100%, with perfect F1-Score, Recall, and Precision, indicating complete class separability after the PCA–LDA transformation.

(ii) For the NSL-KDD dataset, XGBoost achieved the highest accuracy of 92.184%, outperforming other base models with balanced performance across the F1-Score (70.105%), Recall (64.695%), and Precision (94.201%).

(iii) Overall, ensemble and tree-based models, such as Random Forest and Gradient Boosting, demonstrated strong and consistent performance across both datasets, whereas margin-based learners, including SVM and AdaBoost, exhibited comparatively lower generalization under overlapping feature conditions.

This section provides a comprehensive statistical, geometric, and diagnostic evaluation of both individual and ensemble learning models on the UNSW-NB15, NSL-KDD, and NSL-KDD Merge datasets. Quantitative findings from Tables 5.3 and 5.4 are analyzed together with ROC–AUC values and confusion-matrix results to assess model reliability, robustness, and discriminative capacity. In addition, three-dimensional (3D) decision-space visualizations are used to demonstrate class separability and highlight residual zones of misclassification.

All models were trained and evaluated on features transformed through a Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) pipeline. PCA retained between 95–99% of the total variance for UNSW-NB15 and NSL-KDD, while LDA maximized inter-class variance to enhance discriminability. For the merged configuration, both NSL-KDD subsets were combined and repartitioned using a 75:25 train–test split to form a unified dataset, enabling the assessment of generalization under realistic conditions while maintaining comparability across datasets.

5.2.1 Overview of Experimental Configuration

The hybrid PCA + LDA framework was developed to ensure compact yet separable feature representations. PCA reduced redundancy by projecting data onto orthogonal components, while LDA optimized between-class variance and minimized within-class scatter, yielding a discriminative low-dimensional subspace. This combination enhanced computational efficiency and interpretability—both essential for high-dimensional network intrusion data.

Two model families were analyzed. The base classifiers included AdaBoost, Bagging, CatBoost, Gradient Boosting, K-Nearest Neighbors (KNN), Random Forest, Support Vector Machine (SVM), and XGBoost. The ensemble models comprised Hard Voting, Soft Voting, Weighted Voting (Hard and Soft), and a Stacking Classifier. Performance was assessed using Accuracy, Precision, Recall, F1-Score, and ROC–AUC. Confusion matrices provided insights into true and false classifications and the nature of misclassified attacks. For the merged dataset, Accuracy and ROC–AUC were emphasized as indicators of the stability and generalization of learned decision boundaries.

5.2.2 Statistical Performance Analysis

Across all datasets, ensemble and tree-based models consistently outperformed margin-based learners and demonstrated strong generalization. The PCA (95–99%) + LDA transformation produced a stable, near-linearly separable feature space, improving interpretability and robustness.

Table 5.4: Accuracy (%) of Machine Learning and Ensemble Models on UNSW-NB15, NSL-KDD, and NSL-KDD Merge Datasets

ML & Ensemble Model	**UNSW-NB15**	**NSL-KDD**	**NSL-KDD Merge**
AdaBoosting	99.088	86.063	90.282
Bagging Classifier	99.088	88.303	94.676
CatBoosting	99.987	88.267	94.995
Gradient Boosting	99.088	91.812	99.064
K Nearest Neighbors (KNN)	**100.000**	91.443	99.737
Random Forest	99.252	91.789	**99.823**
Support Vector Machine (SVM)	99.424	90.033	98.709
XGBoosting	99.968	**92.184**	99.789
Voting Classifier (Hard)	**100.000**	91.829	99.817
Voting Classifier (Soft)	**100.000**	91.829	99.808
Voting Classifier (Weighted-Hard)	**100.000**	91.825	**99.823**
Voting Classifier (Weighted-Soft)	**100.000**	91.838	99.815
Stacking Classifier	**100.000**	91.896	**99.823**

At a Glance:
(i) For the UNSW-NB15 dataset, the K-Nearest Neighbors (KNN) model and all ensemble classifiers achieved the highest accuracy of 100%, indicating perfect class separability following the PCA–LDA transformation.

(ii) For the NSL-KDD dataset, XGBoost achieved the highest accuracy of 92.184%, demonstrating strong adaptability to overlapping and imbalanced feature distributions.

(iii) For the NSL-KDD Merge configuration, the Random Forest, Weighted-Hard Voting, and Stacking Classifier models jointly achieved the top accuracy of 99.823%, confirming that dataset harmonization and ensemble integration enable near-perfect classification performance.

In the NSL-KDD Merge configuration, inter-class overlap was largely removed, yielding near perfect classification for the top-performing models. The highest accuracies formed a three way tie at 99.823% (Random Forest, Weighted Hard Voting, and Stacking). Other high performers Gradient Boosting (99.064%), KNN (99.737%), XGBoost (99.789%), Voting Hard (99.817%), Voting Soft (99.808%), and Weighted Soft Voting (99.815%) remained within a narrow range below the maximum. Base learners including AdaBoost (90.282%), Bagging (94.676%), CatBoost (94.995%), and SVM (98.709%) showed lower yet stable performance. Collectively, these outcomes supported by ROC–AUC and confusion matrix analy-

ses—demonstrate smoother decision boundaries and minimal false predictions among the ensemble group, confirming that class separability rather than model architecture primarily limits performance on complex data.

For UNSW-NB15, all models achieved exceptional results with 99% variance retention. KNN reached 100% across Accuracy, Precision, Recall, and F1-Score, signifying perfectly separable clusters in the transformed subspace. XGBoost (99.968%), CatBoost (99.987%), and Random Forest (99.252%) displayed strong and consistent results, while Gradient Boosting, Bagging, and AdaBoost each achieved approximately 99.088% Accuracy. All ensemble configurations (Hard, Soft, Weighted, and Stacking) also attained 100% Accuracy. The corresponding confusion matrices revealed no false positives or negatives, confirming complete detection for this dataset.

For NSL-KDD, the PCA (95%) + LDA pipeline revealed moderate variation due to class imbalance and overlapping attacks. XGBoost achieved 92.184% Accuracy (F1 = 70.105%, Recall = 64.695%, Precision = 94.201%), followed by Random Forest (91.789%, F1 = 70.735%, Recall = 65.054%, Precision = 95.808%). Gradient Boosting (91.812%) and KNN (91.443%) also performed well, while SVM (90.033%) and Bagging (88.303%) showed slightly lower generalization. AdaBoost (86.063%) ranked lowest, reflecting its sensitivity to noisy features. Among the ensembles, Stacking achieved 91.896%, Weighted-Soft Voting 91.838%, and Weighted-Hard and Soft Voting 91.825–91.829%. These results confirm the advantage of decision fusion for improved balance and precision. Misclassifications were concentrated in the minority R2L and U2R attacks due to limited training samples and overlapping boundaries.

In the NSL-KDD Merge experiment, performance improved sharply. Random Forest, Weighted-Hard Voting, and Stacking tied at 99.823% Accuracy, followed by Voting Hard (99.817%), Voting Soft (99.808%), Weighted Soft Voting (99.815%), XGBoost (99.789%), KNN (99.737%), and Gradient Boosting (99.064%). SVM (98.709%), CatBoost (94.995%), Bagging (94.676%), and AdaBoost (90.282%) showed moderate improvement under the balanced distribution. Confusion-matrix analyses indicated virtually no misclassifications for the leading models, confirming that the harmonized feature space nearly eliminated inter class overlap and ensured consistent generalization across ensembles.

5.2.3 Misclassification Behavior and Confusion-Matrix Interpretation

Misclassifications primarily occurred within minority attack classes (R2L and U2R) that partially overlapped with Normal or Probe categories. AdaBoost and Bagging exhibited higher false positive rates due to limited boundary adaptability, while Random Forest and XGBoost achieved superior precision and fewer false alarms,

sometimes trading recall for rare classes. In the merged configuration, these errors were nearly eliminated as the balanced data distribution enabled LDA to establish distinct class centroids. Consequently, confusion matrices approached perfect diagonal patterns, demonstrating enhanced robustness and discriminative capability for ensemble models.

5.2.4 Three-Dimensional Decision-Space Visualization

Three dimensional decision space visualizations, derived from the first three PCA components refined by LDA, corroborated the statistical results. The UNSW-NB15 dataset displayed compact, clearly separated clusters for Normal and DoS traffic, consistent with perfect classification. In NSL-KDD, R2L and U2R clusters overlapped slightly with Normal and Probe regions, aligning with confusion matrix observations. The NSL-KDD Merge plots showed dense, well defined clusters with minimal boundary intersections, indicating smooth and stable decision surfaces for ensembles. The alignment of 3D geometry with ROC–AUC and confusion-matrix patterns validates the reliability and consistency of the proposed framework.

5.2.5 Research Implications

The PCA–LDA framework produces compact, variance-rich features that enhance class separation and interpretability. Ensemble learning especially Weighted Voting and Stacking consistently outperforms single models, achieving superior accuracy, high ROC–AUC values, and minimal false detections. Although minority attack classes remain challenging in the original NSL-KDD, the merged dataset mitigates these limitations and yields stable and interpretable confusion structures. The integration of 3D decision space visualization with ROC–AUC analysis further enhances interpretability by providing geometric confirmation of smooth, confident decision boundaries for ensemble models.

Across all datasets, the PCA–LDA approach combined with ensemble learning achieved exceptional accuracy, stability, and interpretability. On UNSW-NB15, KNN and all ensemble methods reached 100% Accuracy, with other models near 99%. On NSL-KDD, ensemble integration particularly Stacking and Weighted Voting improved recall and reduced imbalance driven errors while maintaining Accuracy around 86–92% for single learners and 91–92% for ensembles. On NSL-KDD Merge, Random Forest, Weighted-Hard Voting, and Stacking achieved 99.823% Accuracy, with ROC–AUC values near 1.0, confirming strong generalization and robustness. The combined assessment of ROC–AUC, confusion matrices, and 3D visualization provides a comprehensive understanding of model behavior. The proposed PCA–LDA and ensemble-learning framework thus offers a scalable, explainable, and high-performance solution for intelligent intrusion detection, bridging the gap between benchmark evaluation and real-world deployment.

5.3 Discussion

This thesis integrates quantitative performance evaluation with interactive 3D visualization to enhance the interpretability and robustness of machine learning-based Network Intrusion Detection Systems (NIDS). By combining statistical metrics with visual analytics, this approach bridges the gap between numerical performance and intuitive model understanding, fostering continuous improvement.

The PCA–LDA transformation reduces dimensionality and enhances class separability. For the UNSW-NB15 dataset, the clear geometric structure and high classification accuracy (99–100%) confirm the effectiveness of feature selection. In contrast, the NSL-KDD dataset shows class overlaps, particularly among minority classes, indicating data imbalance and non-linear relationships. The visual decision space helps identify misclassification regions, leading to better feature selection and improved model fairness. Ensemble models, such as XGBoost and Random Forest, outperformed individual learners, especially on rebalanced subsets, improving classification accuracy to 99.823%.

The 3D visualizations support these findings, with UNSW-NB15 showing smooth decision boundaries, while NSL-KDD displays more irregular ones. Ensemble models produced smoother boundaries, reducing overfitting and improving discrimination. The synergy between PCA–LDA and ensemble learning enhances model efficiency, stability, and interpretability, reducing redundancy and improving class separability.

The interactive 3D framework advances model explainability by transforming numerical outcomes into visual insights, enabling analysts to explore misclassification patterns and model confidence. This creates a feedback loop between quantitative metrics and visualization, enhancing transparency and supporting human-guided refinement, in line with Explainable AI (XAI) principles.

This research demonstrates that high-dimensional network data can be effectively visualized and interpreted using PCA–LDA, ensemble learning, and 3D analytics. The proposed framework strikes a balance between interpretability and accuracy, advancing both academic research and applied cybersecurity practices.

CHAPTER VI

FUTURE WORK

Building on the proposed framework, several avenues for future research could further enhance its performance, scalability, and applicability. One promising direction is the exploration of additional dimensionality reduction techniques, such as autoencoders, which could capture more complex, non-linear relationships in the data, thereby improving model interpretability. Enhancing ensemble learning methods and experimenting with hybrid architectures are also critical strategies for improving model performance, particularly in addressing imbalanced datasets and complex feature interactions. Furthermore, integrating threat intelligence from external sources could increase the system's contextual awareness, enabling more informed decision-making, especially in the context of novel or sophisticated attack patterns.

In terms of evaluation, testing the framework on a broader range of real-world datasets, including those from IoT or CICIDS, would assess its robustness and scalability across diverse network environments. Additionally, further advancements in the interactive 3D visualization framework could focus on improving user-centric design to enhance usability and facilitate more intuitive exploration of the decision space. Expanding the customizability of the visualization interface—such as adjusting color schemes, cluster visualizations, and decision boundaries—would allow analysts to tailor the analysis to their specific needs, thereby enhancing its effectiveness.

Moreover, incorporating automated model refinement through active learning techniques would allow the system to dynamically adapt to emerging attack patterns, improving its long-term performance without requiring manual retraining. These future directions will contribute to making the NIDS framework more adaptive, scalable, and explainable, ensuring its effectiveness for real-time, operational deployment in diverse and evolving cybersecurity landscapes.

CHAPTER VII

CONCLUSION

This thesis presents an innovative framework that enhances the interpretability and robustness of Network Intrusion Detection Systems (NIDS) by integrating interactive 3D visualization with machine learning (ML) models. The proposed framework effectively bridges the gap between traditional numeric performance metrics and intuitive model understanding, providing a visual representation of decision boundaries and misclassification patterns in network traffic data. The application of this framework to the UNSW-NB15 and NSL-KDD benchmark datasets demonstrates its efficacy in improving the interpretability of ML models used in NIDS. Specifically, the interactive 3D visualization approach allows for the identification of distinct clusters, such as those formed by Denial of Service (DoS) attacks, while also revealing the challenges associated with classifying mixed traffic. These insights are critical for refining ML models and improving classification accuracy, particularly by visually highlighting areas of overlap and misclassification that are difficult to detect through traditional methods.

Although the results are promising, several limitations related to high dimensional data visualization and the accuracy of decision space representations were identified, primarily due to the dimensionality reduction process and the voxel-based decision space visualization technique employed. Future work will aim to address these limitations by exploring more advanced dimensionality reduction techniques. Additionally, the framework could be extended to handle live data streams, further improving adaptability in dynamic, real-world network environments.

In conclusion, this research contributes to the development of a scalable, explainable, and robust NIDS framework that integrates dimensionality reduction, ensemble learning, and 3D visual analytics. This integrated approach not only enhances the interpretability of ML models but also provides actionable insights that can refine the design and performance of future intrusion detection systems.

REFERENCES

[1] A. Yelizarov and D. Gamayunov, "Visualization of complex attacks and state of attacked network," in *2009 IEEE Workshop on Visualization for Computer Security (VizSec)*, IEEE, 2009, pp. 1–8. DOI: 10.1109/VIZSEC.2009.5375527 [Online]. Available: https://www.researchgate.net/publication/224097772

[2] S. Sangkatsanee, E. Ngan, W. Ooi, and B. Wee, "Real-time intrusion detection system for wireless sensor networks," in *Proceedings of the 2011 International Conference on Computer and Communication Engineering*, IEEE, 2011, pp. 1–5. DOI: 10.1109/ICCCE.2011.6033903

[3] E. M. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys and Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016. DOI: 10.1109/COMST.2015.2494502

[4] J. Camacho, A. Pérez-Villegas, P. García-Teodoro, and G. Maciá-Fernández, "Pca-based multivariate statistical network monitoring for anomaly detection," *Computers and Security*, vol. 59, pp. 118–137, 2016. DOI: 10.1016/j.cose.2016.02.008 [Online]. Available: http://dx.doi.org/10.1016/j.cose.2016.02.008

[5] S. Liu, X. Wang, M. Liu, and J. Zhu, "Towards better analysis of machine learning models: A visual analytics perspective," *Visual Informatics*, vol. 1, no. 1, pp. 48–56, 2017. DOI: 10.1016/j.visinf.2017.01.006 [Online]. Available: http://dx.doi.org/10.1016/j.visinf.2017.01.006

[6] Z. Ruan, Y. Miao, L. Pan, N. Patterson, and J. Zhang, "Visualization of big data security: A case study on the kdd99 cup dataset," *Digital Communications and Networks*, vol. 3, no. 4, pp. 250–259, 2017. DOI: 10.1016/j.dcan.2017.07.004 [Online]. Available: https://doi.org/10.1016/j.dcan.2017.07.004

[7] V. Bulavas, "Investigation of network intrusion detection using data visualization methods," in *Proceedings of the IEEE*, IEEE, 2018, 978-1-7281-0098–2/18/$31.00. DOI: 10.1109/IV.2018.00078 [Online]. Available: https://ieeexplore.ieee.org/document/8412047

[8] M. Ahmed, A. Mahmood, and J. Hu, "Network intrusion detection and comparative analysis using ensemble machine learning and feature selection," *Future Generation Computer Systems*, vol. 92, pp. 348–361, 2019. DOI: 10.1016/j.future.2018.09.057

[9] M. H. Kamarudin, C. Maple, and T. Watson, "Hybrid feature selection technique for intrusion detection system," *International Journal of High Performance Computing and Networking*, vol. 13, no. 2, pp. 232–240, 2019. DOI: 10.1504/IJHPCN.2019.097503 [Online]. Available: https://www.researchgate.net/publication/330660547

[10] W. Zong, Y.-W. Chow, and W. Susilo, "Interactive three-dimensional visualization of network intrusion detection data for machine learning," *Future Generation Computer Systems*, vol. 102, pp. 292–306, 2020. DOI: 10.1016/j.future.2019.07.045 [Online]. Available: https://doi.org/10.1016/j.future.2019.07.045

[11] M. Singh and K. Berwal, "A hybrid method for intrusion detection using svm and k-nn," in *Advances in Computer Science and Information Technology*. Springer, 2021, pp. 177–184. DOI: 10.1007/978-3-030-67187-7_13 [Online]. Available: https://www.researchgate.net/publication/349174383_A_Hybrid_Method_for_Intrusion_Detection_Using_SVM_and_k-NN

[12] Z. Tauscher, Y. Jiang, K. Zhang, J. Wang, and H. Song, "Learning to detect: A data-driven approach for network intrusion detection," *arXiv preprint arXiv:2108.08394*, 2021. [Online]. Available: https://arxiv.org/abs/2108.08394

[13] A. Almomani, M. Alauthman, M. A. Al-Qudah, M. M. Al-Badawi, A. Al-Zoubi, and M. Krichen, "Multi-stage optimized machine learning framework for intrusion detection in iot environment," *arXiv preprint arXiv:2404.13275*, 2024. [Online]. Available: https://arxiv.org/abs/2404.13275

[14] K. Mahesh, *Nsl-kdd dataset*, https://www.kaggle.com/datasets/kiranmahesh/nslkdd?select=kdd_train.csv, Accessed: July 9, 2025., 2024.

[15] M. R. Wells, *Unsw-nb15 dataset*, https://www.kaggle.com/datasets/mrwellsdavid/unsw-nb15?select=UNSW-NB15_1.csv, Accessed: July 9, 2025., 2024.