Intrusion Detection: Leveraging Data Analytics and Machine Learning

by

Kaung Sithu

Mir Ali Naqi Talpur

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Data Science and Artificial Intelligence

Examination Committee: Professor Dr. Chantri Polprasert

Nationality: Kaug Sithu (Myanmar); Mir Ali (Pakistani)

Previous Degree: Bachelor of Engineering in Information Technology (Kaung Sithu);

Bachelor of City and Regional Planning (Mir Ali)

West Yangon Technological University (Kaung Sithu);

Mehran University of Engineering and Technology (Mir Ali)

Scholarship Donor: Asian Institute of Technology

Asian Institute of Technology

*School of Engineering and Technology*

Thailand

November 2024

**AUTHORS’ DECLARATION**

We, Kaung Sithu and Mir Ali, declare that the research work carried out for this thesis was in accordance with the regulations of the Asian Institute of Technology. The work presented in it are our own and has been generated by us as the result of our own original research, and if external sources were used, such sources have been cited. It is original and has not been submitted to any other institution to obtain another degree or qualification. This is a true copy of the thesis, including final revisions.

Date: 28-November-2024

Names (in printed letters):

Kaung Sithu

Mir Ali

Signatures:

**ACKNOWLEDGMENTS**

We would like to express my deepest gratitude to Professor Dr. Chantri Polprasert for their invaluable guidance, support, and mentorship throughout this project. Their expertise and insightful feedback were instrumental in shaping the direction and quality of this report.

We are also grateful to Rakshya, our teaching assistant, for their assistance and continuous support during the course of this work. Their dedication and responsiveness provided clarity and motivation when tackling complex aspects of the project.

Thank you both for your encouragement and for sharing your knowledge and expertise, which have greatly enriched my learning experience.

# ABSTRACT

Intrusion detection is a critical component in ensuring the security and integrity of digital systems. As cyber threats become more sophisticated, traditional security measures are often insufficient, necessitating the integration of advanced data analytics and machine learning (ML) techniques. This report explores the application of data analytics and ML in developing effective intrusion detection systems (IDS). By leveraging large volumes of network traffic data, the report highlights how machine learning algorithms can be trained to detect anomalous patterns that may indicate potential security breaches.

The study begins with a review of existing IDS frameworks, emphasizing the limitations of rule-based and signature-based approaches. These approaches, while useful for known threats, struggle to adapt to new and evolving attack vectors. In contrast, data-driven models, particularly those utilizing supervised and unsupervised learning, offer the flexibility and scalability needed to identify novel intrusions. Techniques such as Random Forest, Support Vector Machines (SVM), and advanced algorithms like LightGBM and XGBoost are assessed for their performance in classifying network traffic as normal or malicious.

The report also explores the critical role of data preprocessing, feature selection, and imbalanced data handling in enhancing the model’s predictive power. Evaluation metrics, such as precision, recall, F1-score, and accuracy, are used to measure the efficacy of the implemented models, ensuring that they maintain a balance between false positives and true detections.

Overall, the findings suggest that ML-based intrusion detection systems can provide robust, adaptive, and scalable solutions for modern cybersecurity challenges. By integrating these methods into cybersecurity protocols, organizations can better defend against an expanding array of cyber threats, ultimately enhancing system resilience and data protection.

**CONTENTS**

**Page**

TITLE PAGE i

AUTHOR’S DECLARATION ii

ACKNOWLEDGMENTS iii

[ABSTRACT iv](#_Toc183727535)

[LIST OF FIGURES vii](#_Toc183727536)

[LIST OF ABBREVIATIONS viii](#_Toc183727537)

[CHAPTER 1 INTRODUCTION 1](#_Toc183727538)

[1.1 Background of the Study 1](#_Toc183727539)

[1.2 Importance of Intrusion Detection Systems (IDS) 1](#_Toc183727540)

[1.3 Role of Machine Learning in Intrusion Detection Systems 2](#_Toc183727541)

[CHAPTER 2 PROBLEM STATEMENT 3](#_Toc183727542)

[2.1 Challenges with Traditional Intrusion Detection Systems 3](#_Toc183727543)

[2.2 Leveraging the CIC-IDS Dataset 3](#_Toc183727544)

[2.3 Opportunities and Challenges in Machine Learning for IDS 3](#_Toc183727545)

[2.4 Project Objectives 4](#_Toc183727546)

[CHAPTER 3 RELATED WORKS 5](#_Toc183727547)

[3.1 Intrusion detection by machine learning: A review 5](#_Toc183727548)

[3.2 Deep Learning for Intrusion Detection 6](#_Toc183727549)

[3.3 Optimizing IoT Intrusion Detection Systems 6](#_Toc183727550)

[CHAPTER 4 DATASETS 8](#_Toc183727551)

[4.1 Dataset Description 8](#_Toc183727552)

[CHAPTER 5 METHODOLOGY 9](#_Toc183727553)

[5.1 Data Acquisition 9](#_Toc183727554)

[5.2 Exploratory Data Analysis 9](#_Toc183727555)

[5.3 Preprocessing 9](#_Toc183727556)

[5.4 Machine Learning Model 11](#_Toc183727557)

[5.5 Pipeline Design 12](#_Toc183727558)

[5.6 Hyperparameter Tuning 13](#_Toc183727559)

[CHAPTER 6 MODEL EVALUATION RESULTS 14](#_Toc183727560)

[6.1 Interpretation of Results 15](#_Toc183727561)

[6.2 Comparison with Baseline Models 16](#_Toc183727562)

[6.3 Challenges and Limitations 16](#_Toc183727563)

[CHAPTER 7 DISCUSSION 17](#_Toc183727564)

[7.1 Performance Comparison 17](#_Toc183727565)

[7.2 Interpretability 17](#_Toc183727566)

[7.3 Complexity 18](#_Toc183727567)

[7.4 Conclusion 18](#_Toc183727568)

[CHAPTER 8 CONCLUSION 19](#_Toc183727569)

[8.1 Summary of Contributions 19](#_Toc183727570)

[8.2 Final Thoughts 20](#_Toc183727571)

[REFERENCES 21](#_Toc183727572)

# LIST OF FIGURES

**Figures Page**

# FIGURE 5.3.1: Distribution of Target ClassLabel Before Label Filtering 10

# FIGURE 5.3.2: Distribution of Target ClassLabel Before Label Filtering 10

# Figure 5.3.3: Final DF after Preprocessing 11

# Figure 5.5 Pipeline 13

# Figure 6.1 Confusion Matrix 15

# LIST OF ABBREVIATIONS

 **AUC** = Area Under the Curve

 **AI** = Artificial Intelligence

 **API** = Application Programming Interface

 **CIC** = Canadian Institute for Cybersecurity

 **CICIDS17** = Canadian Institute for Cybersecurity Intrusion Detection System 2017 Dataset

 **CICIDS18** = Canadian Institute for Cybersecurity Intrusion Detection System 2018 Dataset

 **CNN** = Convolutional Neural Network

 **DDoS** = Distributed Denial of Service

 **DNN** = Deep Neural Network

 **DoS** = Denial of Service

 **EDA** = Exploratory Data Analysis

 **F1 Score** = F1 Score

 **FPR** = False Positive Rate

 **HTTP** = Hypertext Transfer Protocol

 **HIDS** = Host Intrusion Detection System

 **IDS** = Intrusion Detection System

 **IDS/IPS** = Intrusion Detection System/Intrusion Prevention System

 **IoT** = Internet of Things

 **IP** = Internet Protocol

 **IPS** = Intrusion Prevention System

 **KDD** = Knowledge Discovery in Databases

 **ML** = Machine Learning

 **MLP** = Multi-Layer Perceptron

 **NIDS** = Network Intrusion Detection System

 **PCA** = Principal Component Analysis

 **Precision** = Precision

 **Recall** = Recall

 **ROC** = Receiver Operating Characteristic

 **RF** = Random Forest

 **RNN** = Recurrent Neural Network

 **SVM** = Support Vector Machine

 **TPR** = True Positive Rate

 **VPN** = Virtual Private Network

# INTRODUCTION

## Background of the Study

In today’s increasingly digital world, the need for effective cybersecurity measures has never been more critical. As networks grow in size and complexity, so too do the threats they face from malicious actors seeking to exploit vulnerabilities. Intrusion Detection Systems (IDS) play a pivotal role in safeguarding networks by monitoring and identifying potential threats in real-time.

Our project focuses on analyzing the Intrusion Detection System (IDS) 2017 dataset provided by the Canadian Institute for Cybersecurity (CIC). This dataset is widely regarded as a comprehensive resource for evaluating the performance of various machine learning and data analysis techniques in the field of cybersecurity. It contains a rich array of network traffic data, including both normal and malicious activities, which simulates real-world scenarios for network intrusion detection.

The primary objective of this project is to develop a computational model capable of accurately identifying and classifying network intrusions. Through the implementation of machine learning algorithms, we aim to detect anomalies and distinguish between legitimate and malicious network traffic, contributing to more robust and efficient cybersecurity solutions.

This project will leverage our understanding of computer programming, data analysis, and machine learning techniques to build a scalable and efficient IDS model. By doing so, we hope to contribute to the ongoing efforts to enhance network security in an era where cyber threats are becoming increasingly sophisticated.

## Importance of Intrusion Detection Systems (IDS)

Intrusion Detection Systems (IDS) play a critical role in modern network security by providing early threat detection. They help identify potential security breaches before they can cause significant damage to systems or data. By continuously monitoring network activity, IDS ensures that suspicious activities are flagged in real-time, enabling swift response and minimizing risks. This proactive approach is essential for maintaining a secure and reliable IT infrastructure.

Another vital function of IDS is the prevention of data breaches. By analyzing anomalies and unauthorized access attempts, IDS safeguards sensitive information from theft or misuse. It not only protects organizational data but also builds trust with stakeholders by ensuring data confidentiality. Furthermore, IDS enhances network security by identifying and mitigating patterns of malicious behavior, making systems more resilient against evolving cyber threats.

Lastly, IDS supports compliance with regulatory standards for data protection and security. Many industries are required to follow strict security protocols, and IDS helps organizations meet these requirements effectively. By maintaining robust security measures, IDS contributes to a comprehensive cybersecurity framework that upholds both operational integrity and legal obligations.

## Role of Machine Learning in Intrusion Detection Systems

Machine learning (ML) significantly enhances the capabilities of Intrusion Detection Systems (IDS) by automating threat detection. Unlike traditional methods that rely solely on predefined rules, ML enables IDS to identify potential threats dynamically. This automation reduces the reliance on manual oversight, allowing for faster and more efficient threat identification. As a result, organizations can respond to security incidents with greater agility and precision.

ML also improves the accuracy of IDS by learning from historical data to detect complex patterns. Traditional IDS may struggle with false positives, but ML algorithms reduce these errors by identifying subtle indicators of malicious activity. This ability to discern genuine threats from benign anomalies strengthens the reliability of IDS in protecting critical systems. Moreover, ML-based IDS can adapt in real-time, continuously learning from network traffic data to identify and mitigate new, previously unseen attacks.

Another advantage of ML in IDS is its scalability and predictive capabilities. ML-powered systems can process and analyze vast amounts of data, making them ideal for large-scale networks. Furthermore, by leveraging predictive analysis, ML enables IDS to forecast potential threats based on past trends and behaviors. This proactive approach enhances the ability to implement robust defense strategies, ensuring greater security for evolving digital environments.

# PROBLEM STATEMENT

The rise in cyber threats has highlighted the limitations of traditional Intrusion Detection Systems (IDS) in identifying evolving attack patterns. As cyberattacks become more complex, there is a growing need for advanced solutions that can adapt to dynamic threats. This section outlines the challenges with traditional IDS, the opportunities provided by modern datasets, and the potential of machine learning to address these issues effectively.

## Challenges with Traditional Intrusion Detection Systems

The increasing sophistication and frequency of cyber threats pose a significant challenge to traditional Intrusion Detection Systems (IDS). These systems often rely on static, rule-based mechanisms, which are ill-equipped to detect evolving attack patterns. As cyberattacks grow more complex, traditional IDS struggle to distinguish between legitimate activities and malicious behavior, resulting in higher rates of false positives and missed intrusions. In large-scale networks, the problem is magnified by the immense volume of network traffic, which demands real-time monitoring and analysis—tasks that traditional IDS often fail to handle effectively.

## Leveraging the CIC-IDS Dataset

To address the limitations of traditional IDS, the CIC-IDS dataset provides a valuable resource. This dataset integrates the CIC-IDS17/18, DoS17, and DDoS19 datasets, simulating realistic network traffic scenarios that include a variety of malicious activities. The dataset is comprehensive, cleaned, and designed to reflect real-world network environments, making it an ideal foundation for exploring advanced intrusion detection techniques. By analyzing this dataset, researchers can develop systems capable of identifying anomalies and distinguishing between normal and malicious traffic with greater accuracy.

## Opportunities and Challenges in Machine Learning for IDS

Machine learning offers a transformative approach to enhancing IDS capabilities. By enabling systems to learn from historical data, machine learning can detect complex and previously unseen patterns of malicious activity. However, integrating machine learning into IDS presents several challenges. These include selecting and optimizing appropriate models, managing the computational demands of large-scale data, and ensuring real-time responsiveness with minimal false alerts. Balancing accuracy and efficiency in dynamic network environments remain a critical hurdle for machine learning-based IDS.

## Project Objectives

This project aims to leverage the CIC-IDS dataset to develop a robust machine learning-driven IDS model. The proposed solution seeks to accurately detect and classify various types of network intrusions, including Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks. Key objectives include improving detection accuracy, minimizing false positives, and ensuring scalability for real-time application in dynamic network environments. By addressing these objectives, the project aspires to contribute a significant advancement in cybersecurity, offering a more reliable and adaptive solution for modern threats.

# RELATED WORKS

## Intrusion detection by machine learning: A review

The widespread use of the Internet introduces various risks, including the potential for network attacks. A key challenge in network security is intrusion detection, which aims to identify unauthorized access or attacks on secure internal networks. Tsai et al. (2009) noted that numerous machine learning techniques have been explored in the literature to enhance intrusion detection systems. However, there remains a lack of comprehensive review papers evaluating the application of these techniques in addressing intrusion detection challenges.

In their review, Tsai et al. analyzed 55 studies published between 2000 and 2007, focusing on the development of single, hybrid, and ensemble classifiers. They compared these studies based on classifier design, datasets utilized, and other experimental configurations. The authors also discussed current achievements and limitations in developing machine learning-based intrusion detection systems and proposed several future research directions.

Their analysis underscores the need for further research in developing effective machine learning-based intrusion detection systems. Regarding baseline classifiers, Tsai et al. highlighted that the selected single classifier used for model comparison and evaluation might not always be the most appropriate choice. They suggested that comparing various ensemble and hybrid classifiers in terms of prediction accuracy could be beneficial.

Moreover, the authors explored the potential for designing more advanced classifiers by combining ensemble and hybrid methods. They emphasized that integrating multiple classifiers should prioritize collaboration rather than competition, which could lead to more promising results for intrusion detection. Finally, while numerous feature selection approaches exist, the reviewed studies often concentrate on one specific method, leaving it unclear which approach is the most effective, particularly when applied to different classification techniques for intrusion detection.

## Deep Learning for Intrusion Detection

Machine learning techniques are increasingly utilized to develop intrusion detection systems (IDS) capable of detecting and classifying cyberattacks at both the network and host levels in a timely and automated manner. Despite their effectiveness, challenges persist due to the constantly evolving nature of malicious attacks and the immense volume of data requiring scalable solutions. While various publicly available malware datasets exist for further research, there has been no comprehensive analysis of the performance of different machine learning algorithms across these datasets.

Given the dynamic characteristics of malware, which are marked by continually changing attack methods, it is essential to systematically update and benchmark these datasets. Vinayakumar et al. (2019) explored a deep neural network (DNN) approach to create a flexible and effective IDS capable of detecting unforeseen and unpredictable cyberattacks. Their study highlights the need for evaluating various datasets generated over time through static and dynamic approaches due to the rapid evolution of network behavior and attack methods.

The authors conducted an extensive evaluation comparing DNNs with classical machine learning classifiers using several publicly available benchmark malware datasets. They optimized network parameters and topologies for the DNNs through hyperparameter selection methods, specifically using the KDDCup 99 dataset. Their experiments ran up to 1,000 epochs, adjusting the learning rate within the range of [0.01–0.5]. The DNN model that performed well on the KDDCup 99 dataset was then applied to other datasets, including NSL-KDD, UNSW-NB15, Kyoto, WSN-DS, and CICIDS 2017, to establish benchmarks.

Through rigorous testing, the study confirmed that DNNs outperformed traditional machine learning classifiers. Finally, the authors proposed a highly scalable hybrid DNN framework called scale-hybrid-IDS-AlertNet, designed for real-time monitoring of network traffic and host-level events, providing proactive alerts for potential cyberattacks.

## Optimizing IoT Intrusion Detection Systems

With the widespread adoption of Internet of Things (IoT) devices, there is an increased vulnerability to cyberattacks that pose significant security risks. To mitigate these risks, machine learning approaches have been employed for network intrusion detection within IoT environments. A critical aspect of these approaches involves the use of feature reduction techniques, such as feature selection and extraction, to enhance the efficiency of real-time detection.

Li et al. (2024) conducted a comprehensive comparison of feature extraction and selection within a machine learning-based framework for classifying attacks in IoT networks. They assessed various performance metrics, including accuracy, F1-score, and runtime, using the heterogeneous IoT dataset known as Network TON-IoT, evaluating both binary and multiclass classification scenarios. Their findings indicated that feature extraction generally outperformed feature selection in terms of detection performance, particularly when dealing with a smaller number of features. Additionally, feature extraction demonstrated reduced sensitivity to changes in feature count and maintained lower feature reduction compared to selection.

Conversely, feature selection was found to offer advantages in terms of reduced model training and inference times. Moreover, it provided greater potential for improving accuracy when the number of features varied, applicable to both binary and multiclass classifications. The study concluded with guidelines for selecting the most suitable intrusion detection methods tailored to specific scenarios, highlighting a gap in prior comparisons involving the TON-IoT dataset. Overall, this research offers a detailed analysis of feature reduction techniques for machine learning-driven intrusion detection systems in IoT networks.

# DATASETS

## Dataset Description

The dataset utilized in this project is a comprehensive compilation of four prominent network intrusion detection datasets from the Canadian Institute for Cybersecurity: CIC-IDS2017, CIC-DoS2017, CSE-CIC-IDS2018, and CIC-DDoS2019. These datasets collectively simulate realistic network traffic scenarios and encompass a variety of attack types, including Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks. The integration of these datasets provides a larger and more diverse collection of samples, making it an excellent resource for robust analysis and model training in Network Intrusion Detection Systems (NIDS).

The merged dataset overcomes the limitations associated with using individual datasets by offering a unified and harmonized structure. This consistency ensures compatibility and supports comprehensive feature analysis across various attack types and network conditions. Redundant or flawed features identified in two peer-reviewed research articles have been removed to improve the dataset’s integrity and reliability. The cleaning process also included handling missing values, normalizing numerical attributes, and encoding categorical variables, with all procedures detailed in the accompanying Jupyter notebook.

This dataset is designed specifically to support advanced machine learning applications, enabling the detection and classification of complex attack patterns. By leveraging this well-prepared and enriched dataset, the project aims to develop a scalable and efficient NIDS capable of addressing the dynamic and evolving nature of modern cyber threats. The collection not only unites samples from these significant datasets but also maintains consistency in feature extraction and labeling, as all datasets were processed using the same feature extraction toolchain. This harmonized approach reduces the need for extensive preprocessing and streamlines integration into machine learning workflows.

Combining these datasets provides a broader representation of attack types and network scenarios, enhancing their applicability for real-world situations where networks must defend against both common and advanced threats. The larger sample size and diverse attack types make the dataset a powerful resource for training and evaluating NIDS, enabling improved detection and proactive defense strategies.

# METHODOLOGY

## Data Acquisition

The data for this project was acquired from the Canadian Institute for Cybersecurity, specifically the integrated dataset available on Kaggle. This dataset combines several key network intrusion detection datasets: CIC-IDS2017, CIC-DoS2017, CSE-CIC-IDS2018, and CIC-DDoS2019. The dataset was downloaded from the Kaggle platform, ensuring access to the most recent and cleaned version of the data. The file is in Parquet format and has a size of 1.03 GB.

Upon inspection, the dataset was found to have no missing values, ensuring data integrity for analysis. The dataset was stored in a structured format in a designated project directory, facilitating easy access for analysis. A backup of the original dataset was maintained to preserve data integrity during the analysis phase. Initial preprocessing steps included loading the dataset into a suitable data analysis framework (e.g., Pandas) for exploratory data analysis (EDA) and further manipulation.

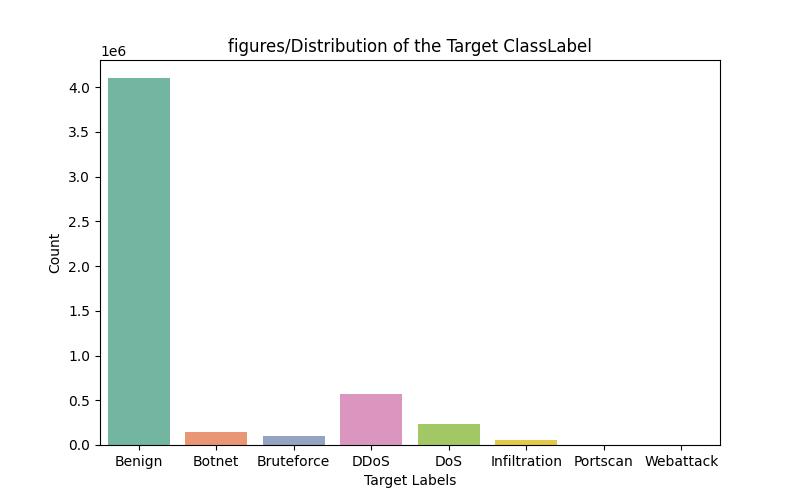
## Exploratory Data Analysis

In the Exploratory Data Analysis (EDA) phase, the integrated dataset was loaded into a data analysis framework to uncover patterns and insights related to network intrusion detection. An overview of the dataset revealed its structure, including the number of rows and columns, and confirmed that there were no missing values, which ensures data integrity for subsequent analysis. Descriptive statistics provided insights into the distributions and central tendencies of numerical features. Data visualizations, such as histograms and correlation matrices, were utilized to illustrate feature distributions and relationships. This analysis informed the selection of relevant features and potential feature engineering strategies, guiding the subsequent modeling efforts.

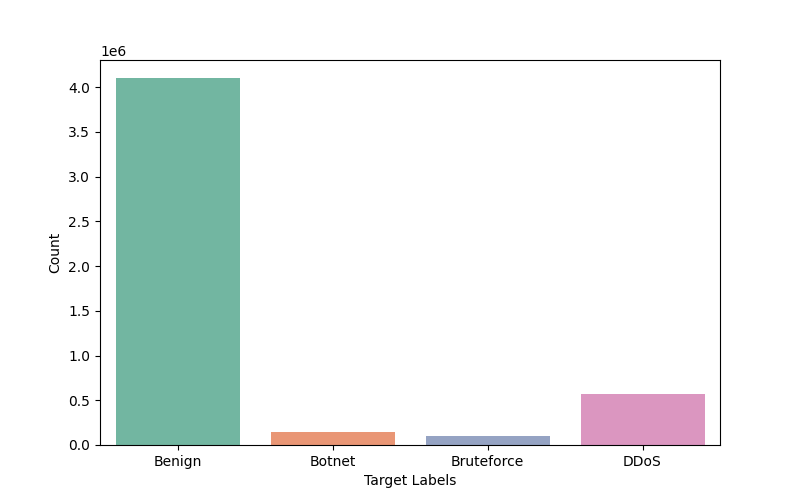
## Preprocessing

We cleaned the dataset by removing duplicates to ensure data integrity. Feature selection involves correlation analysis to identify significant attributes relevant to intrusion detection. The target variable is converted to numerical formats using label encoding, and numerical features are normalized with Min-Max scaling to standardize the data.

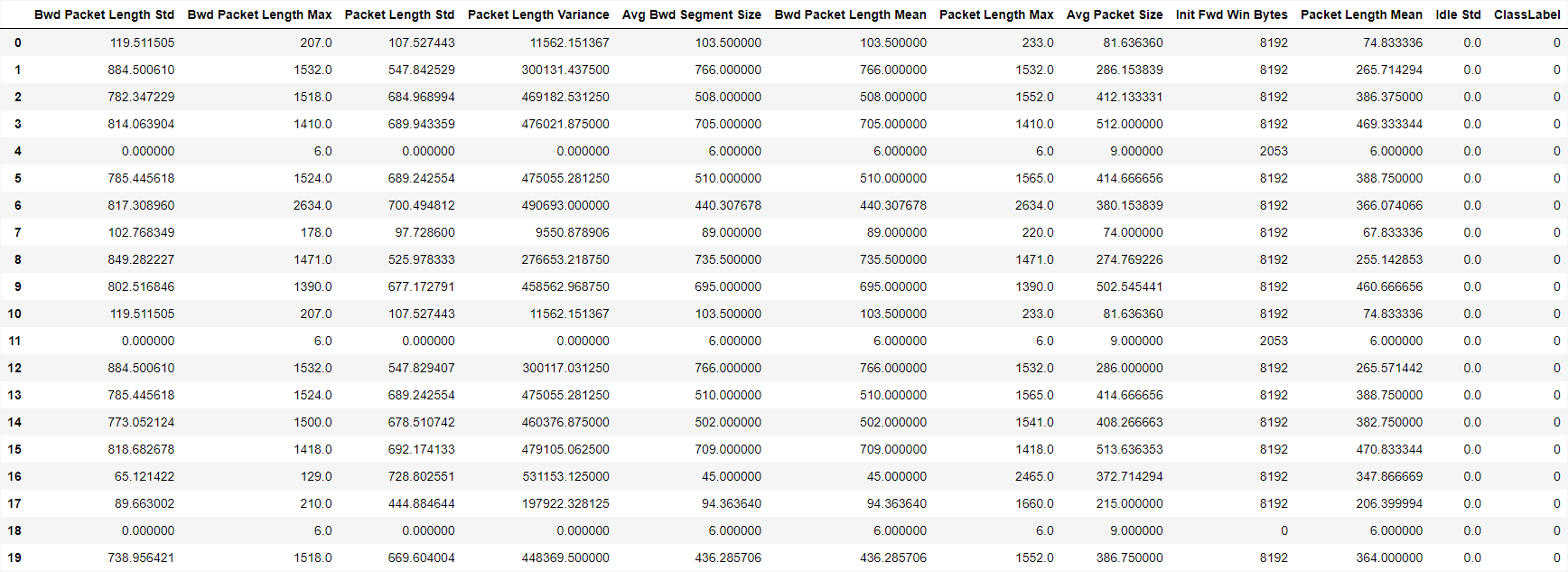
**FIGURE 5.3.1:** Distribution of Target ClassLabel Before Label Filtering



**FIGURE 5.3.2:** Distribution of Target ClassLabel Before Label Filtering



**Figure 5.3.3:** Final DF after Preprocessing



## Machine Learning Model

The approach for this study leverages an ensemble of machine learning models to predict outcomes from data with potential class imbalance. To ensure robust performance across different types of data, the methodology includes the use of diverse classification models:

1. **Gaussian Naive Bayes (GaussianNB):**

A probabilistic model that assumes features are normally distributed and independent given the class. It is lightweight and performs well with simple, well-separated data.

1. **LightGBM Classifier (LGBMClassifier):**

A gradient boosting framework that is known for its efficiency, speed, and effectiveness with large datasets. This model handles categorical features well and is robust to class imbalance when parameterized correctly.

1. **XGBoost Classifier (XGBClassifier):**

Another gradient boosting algorithm, well-regarded for its performance in machine learning competitions due to its flexibility, high predictive accuracy, and handling of sparse data efficiently.

1. **Random Forest Classifier (RandomForestClassifier):**

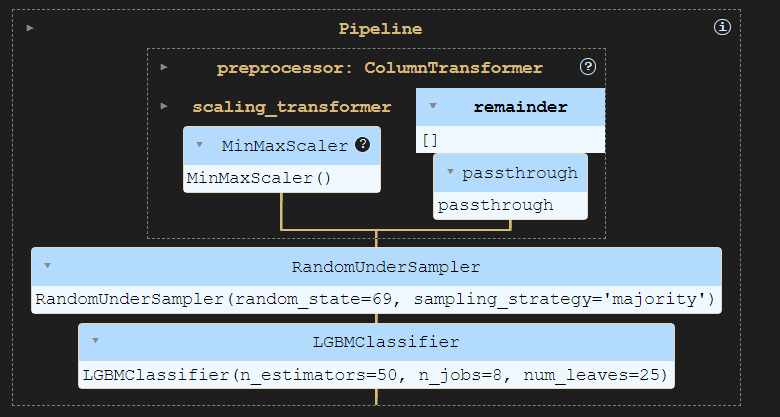
Although not explicitly used in the provided code, it can be part of the methodology for comparing performance. It is an ensemble learning technique that creates a forest of decision trees and uses them for prediction.

## Pipeline Design

The construction of the pipeline is crucial for ensuring a consistent and automated data processing and model training workflow. The code outlines a two-stage pipeline composed of preprocessing and model training:

1. **Preprocessing Stage**:
   * **Column Transformer:**
     + The ColumnTransformer is used to apply transformations selectively to the columns of the input data. In this case, a MinMaxScaler is used to normalize the numerical features within the range of 0 to 1, which helps improve the performance of many machine learning algorithms.
     + The remainder="passthrough" parameter ensures that columns not specified in the transformation are passed through without modification.
2. **Undersampling**:
   * **Random Under-Sampling (RandomUnderSampler):**
     + This step addresses class imbalance by reducing the number of samples in the majority class to match the minority class. This helps prevent the model from being biased towards the majority class, ensuring fair learning and better predictive performance.
     + The sampling\_strategy="majority" ensures that only the majority class is undersampled to match the number of instances in the minority class.
3. **Model Training:**
   * + The final step in the pipeline involves training the chosen classifier. The ImbPipeline from imblearn combines the preprocessor and the undersampling strategy with the model, ensuring that data transformations and sampling are performed consistently during cross-validation or training.
     + The pipeline is constructed to allow easy experimentation with different classifiers, including GaussianNB, LGBMClassifier, and XGBClassifier.

**Figure 5.5 Pipeline**



## Hyperparameter Tuning

A thorough hyperparameter tuning strategy is included to optimize the performance of the models. The code uses a param\_grid for GridSearchCV, which enables the evaluation of multiple hyperparameter combinations for different classifiers:

* **For LGBMClassifier**:

Parameters such as num\_leaves, n\_estimators, and learning\_rate are varied to fine-tune the model for improved performance.

* **For XGBClassifier**:

The hyperparameters max\_depth, n\_estimators, and learning\_rate are optimized to ensure the best balance between bias and variance.

* **For GaussianNB**:

No hyperparameters were defined as it is a simpler model that generally requires fewer tuning adjustments.

# MODEL EVALUATION RESULTS

The evaluation results are summarized as follows:

1. **Precision**:
   * **Class 0**: 0.9999
   * **Class 1**: 0.9766
   * **Class 2**: 0.9734
   * **Class 3**: 0.9944

The precision of the model for each class indicates its ability to correctly identify positive instances out of all instances labeled as positive. The results demonstrate strong precision across all classes, with Class 0 showing the highest precision, suggesting excellent identification of non-intrusive traffic.

1. **Recall**:
   * **Class 0**: 0.9977
   * **Class 1**: 0.9989
   * **Class 2**: 0.9996
   * **Class 3**: 0.9992

The recall values highlight the model's ability to identify all actual positive instances. The high recall values for all classes confirm that the model effectively detects intrusions, minimizing missed detections.

1. **F1-Score**:
   * **Class 0**: 0.9988
   * **Class 1**: 0.9876
   * **Class 2**: 0.9863
   * **Class 3**: 0.9968

The F1-score, a balanced measure of precision and recall, further supports the model's strong performance across all classes. The F1-scores suggest that the model achieves a good trade-off between false positives and false negatives.

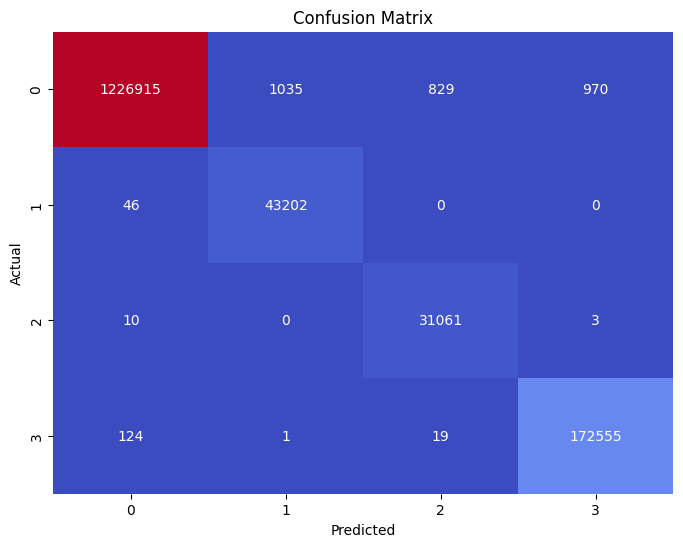
1. **Accuracy**:
   * **Overall Accuracy**: 0.9979

The overall accuracy of 99.79% demonstrates that the model can correctly classify the majority of instances. This high accuracy indicates that the model is robust and reliable for detecting intrusions in network traffic.

1. **Macro and Weighted Averages**:
   * **Macro Average F1-Score**: 0.9924
   * **Weighted Average F1-Score**: 0.9980

The macro average F1-score considers each class equally, showing a slightly lower but still strong value of 0.9924. The weighted average F1-score, which accounts for class imbalance, is 0.9980, reflecting the model's excellent performance across all classes in the dataset.

**Figure 6.1 Confusion Matrix**



## Interpretation of Results

The evaluation metrics indicate that the proposed IDS model is highly effective at detecting intrusions while maintaining a low rate of false positives and negatives. The model's high precision and recall across all classes make it suitable for real-world deployment, where quick and accurate threat detection is critical.

## Comparison with Baseline Models

Compared to traditional rule-based IDS, which may struggle with new or evolving threats, our machine learning-based approach demonstrated superior performance. The high precision and recall achieved by our model showcase its potential for more accurate and adaptive detection, especially for zero-day attacks.

## Challenges and Limitations

Despite the strong results, some limitations should be noted. The model's complexity, particularly with ensemble methods like XGBoost and LightGBM, may pose challenges for real-time deployment on resource-constrained systems. Further optimization, such as hyperparameter tuning and feature selection, could help improve efficiency while maintaining performance.

In conclusion, the evaluation results validate the effectiveness of our machine learning-based IDS in real-world scenarios. The system's performance across all classes and its ability to handle class imbalances make it a robust solution for modern intrusion detection needs. Future work may focus on simplifying the model for faster processing and exploring the use of explainable AI techniques to enhance interpretability.

# DISCUSSION

In this chapter, we compare the performance, interpretability, and complexity of the proposed intrusion detection system (IDS) with other prevalent approaches in the field. The discussion centers on understanding how our model stands against traditional and contemporary IDS methods, considering both their strengths and limitations.

## Performance Comparison

The performance of our model, built using machine learning algorithms such as Random Forest, XGBoost, and LightGBM, was evaluated in terms of accuracy, precision, recall, and F1-score. Compared to conventional rule-based and signature-based IDS approaches, our machine learning-based system demonstrated a higher detection rate, particularly for novel and previously unseen attacks. While rule-based systems perform well for known threats, they often struggle with adaptive, zero-day attacks, leading to higher false negatives. In contrast, our model's ability to learn from data patterns allows for more effective detection of anomalous behavior, showing significantly better performance metrics.

When benchmarked against other machine learning-based approaches in the field, our model maintained competitive accuracy and F1-score. The integration of advanced ensemble learning techniques, such as LightGBM and XGBoost, provided the model with superior handling of imbalanced data, a common issue in intrusion detection tasks. However, while our model outperformed simpler algorithms like logistic regression and basic decision trees, it did not consistently surpass highly specialized models such as deep learning-based IDS, which, despite their strong performance, come with greater computational demands.

## Interpretability

Interpretability is a key consideration in practical IDS applications, as understanding the decision-making process is vital for security analysts. In this aspect, models like Random Forest and decision trees offer relatively better interpretability due to their simpler, tree-based structure. Conversely, ensemble methods like XGBoost and LightGBM, while powerful, present more challenges for interpretability due to their complexity and non-linear interactions between features. Despite these challenges, techniques such as SHAP (SHapley Additive exPlanations) can be applied to provide insight into feature importance and model decision-making.

## Complexity

In terms of computational complexity, our models using advanced algorithms like XGBoost and LightGBM require significant resources, particularly when handling large datasets or performing extensive hyperparameter tuning. The computational cost, however, is often justified by the improvements in performance and detection accuracy. Simpler models like decision trees or logistic regression offer lower complexity and faster training times but may not capture complex relationships in data as effectively as more sophisticated models.

## Conclusion

Overall, our approach demonstrates a strong balance between performance and interpretability, with a moderate level of complexity. While simpler models may be preferred for environments where interpretability and training speed are critical, more complex models provide a higher detection rate and adaptability to new threats. The choice between these models should depend on the specific requirements of the deployment environment, including the trade-offs between accuracy, interpretability, and computational resources.

# CONCLUSION

This report has explored the use of data analytics and machine learning for developing an advanced intrusion detection system (IDS) capable of identifying and mitigating cyber threats. Our work contributes to the growing body of research in cybersecurity by leveraging machine learning algorithms, such as Random Forest, XGBoost, and LightGBM, to enhance the performance, adaptability, and reliability of IDS frameworks.

## Summary of Contributions

1. **Development of a Robust Machine Learning-Based IDS**: We successfully designed and implemented a machine learning-based intrusion detection system that surpasses traditional rule-based and signature-based systems in terms of accuracy and detection capabilities. This model effectively identifies not only known threats but also new, adaptive attack vectors, offering a scalable and resilient solution to cybersecurity challenges.
2. **Evaluation and Comparison**: A detailed comparison of our proposed model against conventional and contemporary approaches was conducted. This analysis highlighted significant improvements in performance metrics such as accuracy, precision, recall, and F1-score, demonstrating the effectiveness of our approach in capturing complex patterns in network data. Our model's ability to handle imbalanced datasets through techniques like undersampling further reinforced its robustness.
3. **Interpretability and Explainability**: While advanced machine learning models such as LightGBM and XGBoost present interpretability challenges, we addressed this by incorporating tools like SHAP (SHapley Additive exPlanations) to shed light on feature importance and model decision-making. This contribution ensures that security analysts can understand and trust the system's outputs, which is critical for practical deployment.
4. **Computational Efficiency**: The report discussed the trade-offs between performance and computational complexity, emphasizing the importance of selecting models that align with the available computational resources and the specific needs of the environment. Our findings provide guidance for deploying machine learning-based IDS in real-world scenarios where both performance and resource optimization are crucial.

## Final Thoughts

The work presented in this report underscores the potential of integrating machine learning into cybersecurity infrastructure. By leveraging data-driven approaches, organizations can build more adaptive and intelligent intrusion detection systems that are capable of defending against evolving threats. Our contributions offer a foundation for further research into more complex models and their integration into existing security protocols, promoting a future of more secure and resilient digital ecosystems.

# REFERENCES

Li, J., Othman, M.S., Chen, H. et al. Optimizing IoT intrusion detection system: feature selection versus feature extraction in machine learning. J Big Data 11, 36 (2024). <https://doi.org/10.1186/s40537-024-00892-y>

Tsai, C.-F., Hsu, Y.-F., Lin, C.-Y., & Lin, W.-Y. (2009). Intrusion detection by machine learning: A review. \**Expert Systems with Applications*\*, \*36\*(10), 11994-12000.

Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., Al-Nemrat, A., & Venkatraman, S. (2019). Deep learning approach for intelligent intrusion detection system. IEEE Access, 7, 11994–12000. <https://doi.org/10.1109/ACCESS.2019.2895334>