A Hybrid Learning-Based Intrusion Detection Framework for Emerging Network Attacks with LIME - Driven Interpretability

by

Md. Mahir Faisal
21301371
Zabia Hossain
21301004
Rahageer Saadman Islam
21201003
Lindsay Prachi Sarkar
21301346
Md. Abdul Kadir
21301733

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

Department of Computer Science and Engineering
Brac University
June 2025

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Md. Mahir Faisal
21301371

Zabia Hossain
21301004

Rahageer Saadman Islam
21201003

Lindsay Prachi Sarkar
21301346

Md. Abdul Kadir
21301733

Approval

The thesis/project titled "A Hybrid Learning-Based Intrusion Detection Framework for Emerging Network Attacks with LIME - Driven Interpretability" submitted by

1. Md. Mahir Faisal (21301371)
2. Zabia Hossain (21301004)
3. Rahageer Saadman Islam (21201003)
4. Lindsay Prachi Sarkar (21301346)
5. Md. Abdul Kadir (21301733)

Of Spring, 2025 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on June 20, 2025

Supervisor:
(Member)

Md. Sabbir Ahmed
Lecturer
Department of Computer Science and Engineering
Brac University

Co-Supervisor:
(Member)

Muhammad Iqbal Hossain, PhD
Associate Professor
Department of Computer Science and Engineering
Brac University

Thesis Coordinator:
(Member)

Md. Golam Rabiul Alam, PhD
Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

The fast-developing artificial intelligence (AI) in cybersecurity has brought the most recent prospects and threats. This thesis examines the weaknesses of AI-based Intrusion Detection Systems (IDS), especially in competitive and adversarial uses that strive to cause model misbehaviors. Starting with conducting a thorough literature review, we are discussing the current methodologies within AI-based IDS and indicating the issues of obscurity of models, scalability, and robustness. Then, a variety of models are applied, ensemble and ordinary machine learning, decision trees, random forests, gradient-based (XGBoost and LightGBM), etc. In order to further promote model reliability and transparency, the Explainable AI (XAI) technique is incorporated, paying particular attention to the LIME (Local Interpretable Model-Agnostic Explanations) method of AI decision-making interpretation. Moreover, we also build and test hybrid ensemble models in order to enhance the accuracy of detection and adversarial resilience. The thesis ends with a demonstration of how explainability and ensemble can be combined to have stronger and more trustworthy and effective intrusion detection frameworks.

Keywords: Intrusion Detection System (IDS), Explainable AI (XAI), LIME, Competitive attacks, Adversarial Threats, XGBoost, Ensemble learning, Cybersecurity.

Table of Contents

Chapter 1

Introduction

The complexity of cyber-attacks continues to rise, which further means that the need for intelligent and adaptive security systems is more important than ever. Intrusion Detection Systems (IDS) can be considered as a very important shield to detect and mitigate possible threats in the networks. The incorporation of Artificial Intelligence (AI), especially Machine Learning (ML) and Deep Learning (DL) has made the IDS work much more advanced as it helps the systems identify new and complex patterns of attacks that are ignored by traditional solutions.

Nonetheless, with the increased use of AI, there have been additional difficulties. Among the major issues, one can distinguish the insufficient transparency of most AI models, which are also called black box systems. Such interpretability is cumbersome to cybersecurity experts because they cannot trust or entirely comprehend the logics of decisions made by these models. Besides, the appearance of competitive attacks when adversaries use or abuse AI models to achieve unfair advantages in detriments to their competitors has also led to the need to develop more resilient and explainable IDS solutions.

To overcome these problems, the current research aims at improving the resilience and visibility of AI-based IDS. Then, we proceed with a systematic literature review to gain insight into the current state of the art of AI-based intrusion detection strategies as well as strengths and weaknesses of the available AI-based methods. Based on these observations, we select and analyze as many different traditional and ensemble machine learning models as possible, such as Random Forest, Logistic Regression, Support Vector Machine (SVM), Naive Bayes, K-Nearest Neighbors (KNN), Decision Tree, and more powerful boosting techniques including XGBoost and LightGBM.

Another important part of our work is the use of Explainable AI (XAI), specifically LIME (Local Interpretable Model-Agnostic Explanations) in interpreting and visualizing model predictions. We hope to augment the integration of LIME in base and hybrid models to bring more transparency of the models and enable security analysts to better grasp the behaviour and judgment of the system.

Also, we are designing and testing hybrid ensemble models where the multiple algorithms are used in order to achieve better overall performance and robustness

against manipulation. These models are not only measured on the basis of accuracy but also on other notable measures of precision, recall, F1-score and false acceptance rate (FAR) to give a comprehensive perspective of the performance of these models in intrusion detection.

By following this research, we would be able to develop an efficient, scalable, interpretable and robust intrusion detection system and provide actionable data to human operators as part of detecting complex attack vectors. This research aids in filling the gap between the success of AI models and explainability and trust in the context of cybersecurity.

1.1 Research Problems

For this research, we have gone through different papers on IDS, and from there, we have pointed out some key problems concerning AI use in cybersecurity. AI models often fail to serve proper instructions for their works, this has made it hard for people who work in related fields to certify correctness or depend on any decisions that are made by these systems because they seem too unclear. Moreover, ensuring the quality of a huge amount of data is also challenging. Also, scalability issues arise when using intrusion detection systems in big IT projects. The complexity and sophistication of cyber threats are on the rise with attackers ranging from script kiddies to nation-states. There is an increasing possibility that AI could expose security loopholes to fraudsters, thereby making their attacks more severe. Additionally, research falls short of discussing adequately how transparency can be achieved in AI models or their performance across diverse datasets thus limiting its generalization and robustness towards different network settings. Finally, due to its dual-use nature, AI has become one of the tools that can be weaponized by cybercriminals hence complicating cybersecurity strategy.

1.2 Research Objectives

The research objectives to mitigate the study's limitations are outlined below:

- The implementation of an explainable AI approaches to improve transparency and interoperability.
- Utilization of edge computing frameworks.
- Design scalable algorithms to solve scalability issues.
- Maintain sophisticated threat intelligence which revers to AI systems that can collect and evaluate threat intelligence regularly to spot trends that point to hackers abusing the technology.
- Modular Integrated Framework that includes interoperability standards as well as designing modular components.
- Implement and describe explainable AI techniques like LIME.

- Use of diverse datasets to ensure model generalizability across different environments.

1.3 Workflow

To research the specified situation, we first done a literature review to learn the existing tendencies in terms of the Intrusion Detection Systems based on AI, as well as issues that emerge due to competitive and adversarial attacks. Depending on what we learned, we discussed alternative methodologies that can be applied to overcome these challenges. Then we simply tested a range of machine learning models, both traditional and ensemble, and enhanced boosting models, Random Forest, XGBoost, and LightGBM. We assessed performance of individual models and then we created and tested some hybrid ensemble methods to improve detection accuracy and robustness. At the last stage, we have used Explainable AI methods, especially LIME, to explain model predictions and help make them more transparent. The systematic workflow process has helped us in this study to develop a better and explainable intrusion detection system framework.

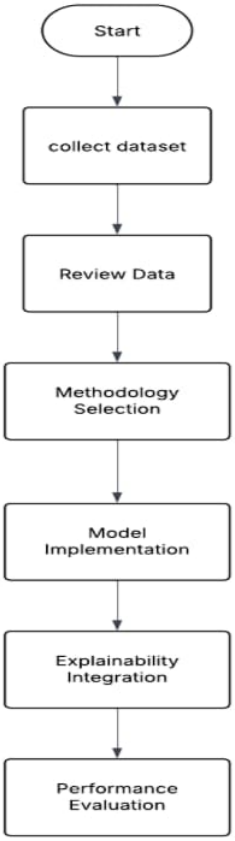

Figure 1.1: Workflow Diagram

1.4 Thesis Structure

This thesis is organized into seven chapters. Chapter 1 introduces the research background, problem statements, research objectives, and the workflow of the entire study. Chapter 2 establishes the most detailed literature review which examines the historical development of intrusion detection systems (IDS), development of artificial intelligence (AI) in cybersecurity, and the competitive and adversarial attacks challenges. Chapter 3 explains the dataset that was employed in this study, that is, the dataset attributes, the occurrence of classes, features relevancy, and how the dataset was preprocessed, and ways to avoid data breach. Chapter 4 is dedicated to the application of various machine learning models including Decision Tree, Random Forest, Naive Bayes, Gradient Boosting, XGBoost, and LightGBM and their mathematical models and references to scientific researches. Chapter 5 measures the accuracy of all the models that are run by standard measures, such as accuracy, F1-score, and ROC AUC and proposes the use of hybrid ensemble models as the purpose of achieving greater accuracy and robustness. Chapter 6 will use Lime architecture and explainability analysis to deliver visual, explainable, interpretable model predictions, to allow the user to understand what features contribute to particular decisions. Lastly, Chapter 7 summarizes this thesis by compiling the primary inferences, ideas, and possible future studies.

Chapter 2

Literature Review

Sowmya et al. (2023) represented the intrusion detection mechanisms, comparing their performance compared to traditional security methods. The detection mechanism consists of parameterization, training, and detection. The authors gave a demonstration of how the IDS methodology can be classified based on efficacy, precision, and performance. AI-based mechanisms showed the best detection and classification performance with feature reduction being a mandatory step. Many new procedures were presented by the authors, including the Support vector Machine (SVM) and KNN algorithm for supervised ML, Fuzzy C means clustering and k-mean clustering for unsupervised ML, Artificial Neural Networks, Convolutional Neural Networks, and Recurrent Neural Networks for supervised deep learning, and Auto Encoder and deep Belief Networks for unsupervised deep learning. Performance of above 99 percent were comparable for ML, DL, and ensemble-based techniques. From the collected dataset, the findings show that ensemble-based IDS was successful in identifying network threats and that DL-based IDS outperformed ML-based IDS in terms of accuracy rates [23]. The study only deals with common security attacks and concentrates on accuracy metrics. This research attempts to describe several AI-based intrusion detection schemes and give more insights to understand better in the future, the challenges that researchers may face while trying to identify multi-class attacks.

Zeadally, S. (2020) In this study the authors have examined the changing field of cybersecurity, focusing on the increasing complexity and economic nature of cyberattacks. Cyber security threats can originate from a range of sources, including script kiddies, criminal organizations, nation-states, terrorists, spies, disgruntled employees, and external and insider attackers/threats. To mitigate these risks, traditional (non-AI) cybersecurity strategies have been employed, including game theory, rate control, heuristics, signature-based intrusion detection, anomaly based intrusion detection, end-user security control, and autonomous systems. The implementation of artificial intelligence (AI) techniques especially, machine learning algorithms—to improve cybersecurity measures is explored in detail in this paper. The applicability of several machine learning techniques, such as decision trees, supervised learning, artificial neural networks (ANNs), k nearest neighbors (k-NN), support vector machines (SVMs), and self-organizing maps. The use of AI to improve cybersecurity in a variety of fields, with a particular emphasis on the Internet, the Internet of Things (IoT), and critical infrastructure, is also explained here. To conclude, the

study mainly emphasized the importance of AI in cybersecurity which also notes how essential it has grown as cyber threats get more sophisticated [10].

Tcydenova and Kim (2021) represented significant vulnerabilities for adversary attacks representing intrusion detection systems (IDSs) created by machine learning. This was demonstrated by running FGSM and PGD attacks against different models, resulting in accuracy dropping from 0.750 to 0.5. Similarly, alternative model attacks including GAN and ZOO reduced the DNN-based IDS accuracy from 0.8896 to 0.5. It is clear from this research that when trained with IDS, concepts of adverse instances can be identified that effectively contribute to improving safety potential. This paper focuses on studying adversarial attacks on DNNs through XAI. This gives an insight into how a SHAP-based approach was developed to improve detection accuracy and reveal model weaknesses. To interpret and verify the decisions made, IDS incorporates XAI into the framework of IDS using SVM classifiers. These efforts are essential to building resilient, transparent IDSs capable of resisting sophisticated adversary attacks [13].

Poudyal, S. (2020) presented an AI-powered ransomware detection system known as AIRaD. Their tool combines static as well as dynamic analysis using such tools as Cuckoo Sandbox, PIN, Ghidra, NLP, and association rule mining. Their approach has 99.54 percent accuracy for identifying ransomware along with low false positive rates. They follow an approach that has hybrid reverse engineering, multi-level code analysis,ML processing, ML classification and ransomware signature database construction. Its development on a dataset which contains 550 samples of ransomware has great accuracy but very low false positives in experiments. As it is easy to understand and comes with a machine learning method that supports people while tracking down viruses, AIRaD is proposed open source by them. This tool shows how important chains of behavior and file encryption are for data set construction and model training using machine learning. Through their endeavors, they developed a framework called Automated Inverse Reverse Engineering Driven Framework(AIRaD) which automatically reverse engineers software looking at the role of behavioral chains in ransomware detection. Finally, this research paper has introduced AIRaD architecture which is better than other traditional methods used to detect ransomware [9].

Sriram et al (2022) demonstrated the application of machine learning and deep learning procedures in intrusion detection systems. The authors focused on anomaly based IDS as they have the capability to identify undetermined attacks by detecting abnormalities from normal network behavior. The paper also highlights the crucial role of explainability in AI models that are preferred by IDS. Some Explainable AI tools such as Shapley Additive exPlanations (SHAP) and Local Interpretable Model-agnostic Explanations (LIME) are featured because of their contributions in the improvement of ML and DL models' transparency. The study also examines various ML and DL approaches like Random Forests (RF), K-Nearest Neighbors (KNN), Decision Trees (DT) and Support Vector Machines (SVM), which are used in IDS for their role in categorizing normal and malicious activities. Moreover, a discussion about the use of Principal Component Analysis (PCA) is presented since it has the capability to improve the accuracy of detection by reducing data dimen-

sionality. Lastly, the future guidelines for research have suggestions like exploration of techniques like DeepLIFT, which could offer more explanation accuracy and efficiency for DL models in comparison with recent methods [19].

In this article, Tatineni (2024) conducts a study that investigates how computers could be trained to identify anomalies in cybersecurity. SVMs, Neural Networks, and Ensemble Methods are some of the techniques they review. The last two have been tested and proven effective for rapidly adjusting to new changes and are, also, they can operate at any rate hence ensuring threats in computer systems are within their reach. Despite this though, the processes behind them remain unclear as well as cyber-attacks from hackers which can tamper with innocent individuals' personal data and information security among other things, making us increasingly vulnerable. The author suggests that more research as well as collaboration is needed to enhance security. By examining various computer approaches, the study offers practical tips for those striving to secure PC systems from risks. Among such individuals are computer users, theorists, and decision makers. On the contrary, although the paper does not cite the exact datasets, it clarifies on how they can apply this method in the act to ensure safety in computer machines [12].

In their article, Him et al (2024) sought to make better proactive threat detection and incident response by integrating artificial intelligence (AI) as well as machine learning (ML) into real-time threat intelligence frameworks. Large-scale data sources are analyzed, trends are extracted, and ever evolving datasets are continuously fed into AI powered systems. No more repeating the same task over time once the machines can do it instead. Adversarial evasion techniques, algorithmic bias, and data privacy are among the difficulties. Cybersecurity paradigms that stress reactive measures may be exploited by stealthy attacks against businesses. To deal with cyberattacks, proactive threat intelligence systems use AI and ML can analyse massive volumes of data in real-time. Real-time threat analysis includes Real-time Stream Processing, Dynamic Risk Assessment using AI, Predictive Analytics for Threat Identification. This paper aims at exploring theoretical frameworks, ways of analyzing data and practical features of application of AI/ ML in real-time threat intelligence systems for enhancing cybersecurity capabilities [25].

Dash, B. (2022) The authors of this paper have looked at how artificial intelligence (AI) is changing several industries, with cybersecurity being one of its main benefits due to its ability to build strong defenses against cybercrime. Profit-driven cybercriminals employ techniques including phishing, DoS assaults, and cyber espionage. Cybercriminals have also used AI to launch complex attacks, which has made the need for sophisticated anti-phishing algorithms necessary. The serious financial losses and threats to national security that result from cybercrime underscore the urgent need for efficient countermeasures. Cybersecurity is becoming much more dependent on artificial intelligence (AI), which improves data protection via technology, detection, prediction, and quick response [14]. Artificial intelligence's predictive powers, fueled by machine learning, allow for examining massive data sets to anticipate possible cyber threats. The possibility for hackers to abuse AI is a major worry since they can utilize AI systems to find weaknesses and intensify their attacks. For the effectiveness of Ai-based cybersecurity, the accurate models

determined by precession and efficiency are required. These measures help in the assessment of the accuracy and responsiveness of an AI model in detecting threats. The inclusion of AI in cybersecurity also has many challenges, such as high investment rate, low rate of awareness and also the risk of cybercriminals abusing the models. Despite having difficulties, AI is automated and error free. For this reason, it can effectively prevent attacks and enhance defense systems. The advancement of cybersecurity defenses is also important along with the threats and also leveraging the ability of AI to predict, detect, and respond.

Jakkka et al. (2022) provide a thorough overview of identifying malware and managing cyber risk using Artificial Intelligence (AI). According to the paper, AI minimizes risks such as malware attacks and cyber crime while enhancing security of organizations' IT systems. Besides examining how companies are experiencing increased incidences of malware, there is a comparison between AI-driven cybersecurity initiatives with those from other sectors. It combines 50 questionnaires for IT experts and extensive review of relevant literature to offer insights on how AI detects and mitigates cyber threats. They conclude that AI complements humans in securing cyberspace but is also faced with challenges like biometric data vulnerabilities, costliness when implemented, and susceptibility to adversarial manipulation [16]. However, behind these threats lies the potentiality of AI in terms of identifying unknown threats or providing foolproof authentication making it relevant in today's network defense domain. The future work suggestions include developing AI implementations that are cost efficient and also address privacy concerns regarding the use of biometric data.

Manninen, n.d. proposed how artificial intelligence (AI) can be inserted into Intrusion Detection Systems (IDS) for the purpose of enhancing their effectiveness. The author identifies various aspects that deal with AI integration with IDS such as types of learning processes used, detection principles applied, and response to noise [27]. According to Manninen, the ability of AI-based systems to learn continuously is crucial for the machine learning process where it addresses issues such as reducing human intervention and increasing system accuracy. Traditional IDS are depicted by Manninen as limited when compared to those that are based on AI which uses examples rather than set rules, which he explains. Using fuzzy logic, neural networks, and genetic algorithms are some of the AI systems that can support IDSs to improve accuracy as well as speed. The soundness of a thorough investigation regarding the noise during training is a must when it comes to this aspect of intrusion detection and protection. The concluding part of his extensive review includes salient points that outline AI's present condition and potential trends supported intrusion detection systems (IDSs) by Manninen who seems to indicate that they could revolutionize network security procedures.

The paper by Rawal (2022) [18] is about Explainable AI (XAI), which makes AI transparent and reliable. They explain its development, objectives as well as security while comparing ways of making it clear. The problem with AI/ML systems is that they are indispensable but raise questions about bias. The research and regulations of XAI strive to ensure just decisions that can be relied upon. It is crucial for non-experts to understand how AI arrives at its decision premises. This

confidence helps users in questioning the results more effectively. XAI tackles fears about complicated deep learning models by simplifying their choices and behaviors.

Owen et al (2023) [21] review in their study examining the difficulties associated with adversarial machine learning in cybersecurity, with a particular emphasis on model extraction, poisoning, and evasion strategies. It therefore advocates for a complex defensive strategy that employs AI models which are robust, resilient, and adaptable. This paper also discusses the intersections between traditional cyber security concepts vis-a-vis adversarial machine learning. In AML (adversarial machine learning) which is in the expanding field of cybersecurity with AI, attackers who use maliciously manipulate AI systems due to their vulnerabilities. The trustworthiness and efficiency of AI-driven solutions are greatly affected by this. Some examples include Gradient-Based Attacks, Optimization-Based Attacks, Transferability Attacks, White-box vs. Black-box attacks, and physical Attacks among others were highlighted by the author. By analyzing these kinds of attacks one can create strategies like Feature Space Transformation; Input Preprocessing Methods; Defensive Distillation; Ensemble and Defensive Model Stacking to counter them respectively. To further develop adversarial machine learning through cybersecurity and overcome challenges as well as defeat attacks and navigate moral and legal dilemmas, collaboration among academia, the business community, government bodies as well as civil society is paramount. The article explores adversarial machine learning within cyber security highlighting its wide range of application areas as well as how strong defenses are needed hence further research should focus on emerging issues. Furthermore, the strength of a robust artificial intelligence system lies in risk mitigation; maintaining operations; and conserving resources in an era characterized by digital technology.

Schmitt, M. (2023, December) The relevance of AI in achieving cyber resilience is emphasized in this research as it focuses on the resilience of anomaly-based intrusion detection systems offered by AI and the difficulties in integrating them into intricate IT infrastructures. IDS are classified into two types: signature-based and anomaly-based. Their goal is to automatically identify cyberattacks. To solve integration issues across process infrastructure dimensions, and data application, this research focuses on integrating AI-based intrusion detection systems into corporate systems. To improve AI-based intrusion detection and handle integration issues in complex digital settings, the study places a strong emphasis on accuracy, resilience, and robustness. With an emphasis on resolving integration issues, the research uses AI-enabled cybersecurity models to safeguard contemporary digital ecosystems. The study looks at IoT, mobile, and network security. The report also gives a general review of several kinds of cyberattacks, especially those that are pertinent to IoT ecosystems, like malware, DDoS, injection, and MITM attacks. The discussion then moves to the concept of "digital ecosystems," which combine AI, IoT, and other technologies to build intelligent, networked systems. Researchers deliver the numerical outcomes of threat detection scenarios, incorporating IoT, malware on Android devices, and network penetration. Moreover, there are difficulties with integrating AI/ML-based security solutions into intricate digital ecosystems, which emphasizes the necessity for methods like Industrial Information Integration Engineering (IIIE) to make integration and deployment easier. Integrating AI-enabled cybersecurity

solutions presents particular problems in the areas of network security, mobile security, and IoT/CPS security [22].

Li et al. (2022) created a concrete security strategy driven by artificial intelligence which would anticipate and protect from advanced persistent threats (APTs) in remote networks, involving such approaches as Explainable Artificial Intelligence (XAI), Cyber Threat Intelligence (CTI), and game theory. This may expose edge devices with limited resources to great danger because these devices are characterized by sustained delays from infected APTs, highly targeted impacts from them, and indeterminate actions quantifying possibilities. They deployed an edge defense mechanism for countering APT attacks with immediate response to mitigation using XAI as a means of AI decision explanations and CTI as a trusted threat intelligence generator . In order to achieve quick response time and ensure effective detection with resource optimization, this plan has been built around DRL tactics. The result from the series of experiments shows that proactive learning and high-accuracy detection enhances the protection level and defense capability of the edge devices against APT [17].

In their research paper, Muneer et al. (2024) have given a proper description of artificial intelligence (AI) in intrusion detection (ID) systems. The authors' paper puts its focus on DL's ability to automatically learn features as well as highlights the need for vast labeled data and computing resources. Nevertheless, ML techniques which are less computationally intensive struggle with generalizing to new data. On the other hand, FL is seen as a promising approach towards collaborative learning but with increased communication overheads whereas XAI enhances model transparency. This comprehensive review is aimed at assisting practitioners by examining the strengths, weaknesses, and applicability of these techniques under different network settings and resource limitations thus filling significant gaps in the present literature on context-based ID applications [26].

Sharfuddin et al (2020) represented five machine learning algorithms using the UNSWNB15 dataset in their study. According to him, Random Forest is proven to be the best classifier based on metrics like false positive rate, F1 score, and detection accuracy. However, as computer networks and applications grow, additional obstacles are experienced by cybersecurity. Some examples of modern security measures include encryption and firewalls but none of them is perfect. Network Intrusion Detection Systems (NIDS) observe network traffic flow in order to identify any dangerous activity. This technology boosts their performance by training IDS algorithms using datasets based on network security. The suggested technique extracts features from the UNSW-NB15 dataset, separates it into training and test sets, trains five machine learning classifiers, and selects the most accurate model for predicting benign or harmful data presence. The suggested system utilized Random forest, Ada Boost, Decision Tree, Gradient Boosting, and Gaussian Naive Bayes algorithm. With 98.6 percent accuracy on the UNSW-NB15 dataset, Random Forest stands out as a top machine learning classifier indicating its potential for intrusion detection purposes [7].

HUH, S. (2022) represented in the study investigates malware propagation and eva-

sion strategies by gathering malware binaries and URLs in real-time data. New research on IoT malware has shown correlations with conventional malware. The work conducted by the authors sets itself apart by employing real-time data collection techniques to conduct an extensive analysis of malware binaries and dissemination URLs.The accuracy and efficiency of automated malware analysis are improved by machine learning and deep learning. Over 287 days, the data collecting strategy entails obtaining malware distribution URLs and binaries from publicly accessible sites such as URLHaus and VxVault. These are gathered in real time by a specialized crawler that periodically checks sources to guarantee current data. Variations in malware family classification by antivirus manufacturers are revealed by analyzing binary samples of malware that have been gathered. Based on file similarities, malware binaries could be effectively clustered, yielding 868 clusters containing 44,104 samples, mostly Trickbot, Mirai, and Emotet. By using this strategy, malware classification algorithms can be improved by detecting unidentified no samples that need inspection by hand and prioritizing analysis accordingly. It intends to give users real-time insights and promote community contributions for better detection with plans for ongoing improvements. Trends in the dissemination of malware show an increase in URLs over time, with the average uptime of malicious and benign websites varying. Our research's conclusions emphasize the necessity of uniform malware family classification, cross-platform malware analysis tools, and preventative actions against adversaries' changing tactics. Malware defense will be strengthened by initiatives to create standards for classification, improve distribution site detection, and develop cross-platform analytic approaches [15].

Kumar et al. (2010) propose a comprehensive review on the use of Ai techniques for intrusion detection system (IDS). The basis of categorization of IDS in this paper is influenced by diverse features such as source of audit data, processing methods and response types. The article discusses many approaches, including Decision Trees, SVM, NN, Data Mining, Genetic algorithms, Fuzzy Logic, Bayesian Networks, Markov models, and clustering algorithms. This study examines various methods: a comparison of effectiveness when considering different parameters like dataset used, feature reduction techniques and classifier design have been made to each other. There are three main groups considered: a single technique-based IDS, hybrid or ensemble IDPS using multiple classification techniques . In conclusion, the authors pointed out the pros and cons of AI-based intrusion detection systems with an emphasis on their flexibility and adaptability, pattern recognition capability, and computational efficiency. They also highlighted areas that require further research to improve the effectiveness of AI on intrusion detection [1].

Fathia, A. (2023) analyzed some weaknesses in AI-based cybersecurity systems that make them prone to adversarial attacks. Towards this, the author suggests a taxonomy of attack approaches such as evasion, poisoning and data injection. Simultaneously, this research considers traditional mitigation techniques like making models resilient to onslaughts or purifying inputs for future investigations; it includes Adversarial Training, Ensemble Methods and Anomaly Detection. From this essay also there are decent recommendations on interdisciplinary approaches such as game theory and cognitive security for better resistance against complex threats. The author also touches on ethical issues and societal implications of adversarial machine learn-

ing. Thus, according to the survey, AI-powered cybersecurity systems must have an active defense mechanism at all times. Also, the writer anticipates future research avenues like securing cyber machines through man-machine cooperation too. In doing so, one has to interpret constraints in terms of improving data quality while minimizing falsifications from adversaries at the lowest possible cost with efficient cost-effective but scalable solutions that reduce big dataset manipulations through interventions by bad actors [20].

Aljabri et al. (2021) gave a comprehensive overview of the latest intelligent techniques in network attack detection. These authors investigated ML, approaches, DL, and their efficiency and limitations in addressing various network threats. Also, this article presents some examples of previous techniques such as port-based solutions and payload base approaches that had become clever. Lastly, this review stresses on the importance of strong training data sets, good algorithms and dependable performance measures to improve detection accuracy. The author extensively discusses intelligent detection techniques categorization, network attacks emphasizing more on adaptive systems that can be used to handle emerging threats. In future, it is foreseen that stronger models will be developed that can manage encrypted data, reduce false positives and integrate multiple security layers. Further still, there are prospects for more research on hybrid models which combine traditional ones with DL or ML towards enhancing overall security frameworks. This research work provides a valuable resource for students/educators who would like to improve their networks [11].

Chapter 3

Dataset Description and Preprocessing

3.1 Dataset Overview

The research is conducted on the basis of the sub-dataset network traffic of the TON_IoT dataset, which is provided by UNSW Canberra [8]. TON_IoT framework is a common benchmarking metric exploited in the assessment of intrusion detection ventures (IDS) in contemporary cyber-physical settings. The sub-set which is used in this work is named as big_network_dataset_1 and the number of samples in it is 97,250 and the attributes are 46 in total, which covers wide and realistic aspects of network behaviors and attacks involved.

The dataset consists of features at the low level (destination/source IP addresses, port numbers and type of protocols) and those at the high level (attributes at the application layer, especially those of HTTP and DNS headers). They are also complemented by the statistical measurements, which include packet ratios, number of bytes and time intervals. All the features combine to provide an all-encompassing picture of network activity and by extension, the dataset is very appropriate to use in training and testing machine learning models within the intrusion detection framework.

The structure of the data is 29 categorical (nominal) variables, 16 integer fields and 1 floating-point attribute, resulting in a high-dimensional and heterogeneous feature space. This diversity enables the models to adjust to the symbolic-based protocols and numeric flow-centered behaviors. The dataset enables detection of a broad range of threats by covering many layers of the network stack, such as DDoS, injection, scanning, XSS and password attacks as well as benign traffic.

The network of this subset of TON_IoT is especially appropriate to test hybrid ensemble models and explainability frameworks because it exhibits more realistic and more realistic network interactions as they appear in practice at the enterprise or industry level.

3.2 Target Variable and Class Distribution

In this case, the target (label) variable defines each network flow as one of six classes:

- Distributed Denial of Service (DDoS)
- Injection attack
- Normal (benign traffic)
- Password attack
- Scanning activity
- Cross-site scripting (XSS)

The classes are roughly equally represented, with the largest proportion of 18.7 percent and the smallest one of 14.2 percent, which does not make the classification task severely unbalanced. These equally distributed classes are preferable to train supervised learning models, since they are less biased towards majority classes and improve the model's generalization capability.

3.3 Data Integrity and Feature Characteristics

Extensive integrity test run on the dataset revealed no missing values in important columns like timestamps, source and destination IPs, port numbers, service identifiers, and byte-level measures. But 47 duplicates were found and they were deleted to make the training set pure. Some of the categorical attributes had high cardinality, such as source bytes, destination IP addresses, and DNS query types. Such characteristics had to be thoroughly encoded and dimension reduced in the preprocessing step to avoid overfitting.

Feature	**Description**	**Role in Intrusion Detection**
ts	Timestamp	Capturing time. Assists in identifying the patterns of attack timing.
src_ip	Source IP Address	Capturing time. Assists in identifying the patterns of attack timing.
src_port	Source Port	Source port. Assists in scanning of ports or service access.
dst_ip	Destination IP	Traffic target. Important to determine attack destinations.
dst_port	Destination Port	Receiving port. Target service may be displayed in ports such as 22 (SSH), 80 (HTTP).

proto	Protocol	Transport protocol (UDP/ICMP/TCP). It is utilized to derive the behavior of traffic.
service	Application service	HTTP, FTP detected. Linked with app weaknesses.
Duration	Connection duration	Unexpected long or short durations, possibly shell flooding or incomplete attacks.
src_bytes	Bytes sent from source	Outgoing big data could mean exfiltration or DDoS.
dst_bytes	Bytes sent to destination	Size of the inbound data; applicable in downloads or responses.
conn_state	Connection state	e.g. S0 (SYN sent, no reply). Identifies scans or failure.
missed_bytes	Undelivered bytes	Signifies that packet drop may have occurred, or tampering may have occurred.
src_pkts	Packets from source	Initiators packet count. Useful to spot floods.
src_ip_bytes	Total bytes from src IP	Amount of aggregated data. Useful in the profiling of hosts.
dst_pkts	Packets from destination	Response traffic. Reflections attacks can be detected as anomalies.
dst_ip_bytes	Total bytes to dst IP	Incoming data in total. Presents download patterns.
dns_query	DNS Query Name	Querying domain. Is able to identify DGA or malicious domain.
dns_qclass	DNS Query Class	The acronym is usually IN (Internet). Abnormalities can be reflected at unusual values.
dns_qtype	DNS Query Type	A, AAAA, MX, and so on. These are used to realize the nature of the DNS requests.
dns_rcode	DNS Response Code	0 means no error; any other value can be failure or poisoning.
dns_AA	Authoritative Answer flag	True/False. Refers to authority DNS reply.
dns_RD	Recursion Desired	True when the client wishes a recursive solution.
dns_RA	Recursion Available	Stipulated by DNS server in the event that recursion support is enabled.
dns_rejected	Query rejected by server	Boolean flag; this can be used to suggest policy-based denial.
ssl_version	SSL/TLS Version	Aids discovery of weak/old encryption protocol.

ssl_cipher	Cipher suite used	Poor cipher suites depict threats.
ssl_resumed	Session resumed	True/False. Could impact the security of sessions.
ssl_established	Was SSL established?	True/False. Identifies handshakes failures.
ssl_subject	SSL certificate subject	It specifies the owner of the certificate.
ssl_issuer	SSL certificate issuer	CA which issued the cert. Assists in identifying false certs.
http_trans_depth	HTTP transaction depth	Quantity of inner requests. Great depth could be odd.
http_method	HTTP request method	GET, POST, etc. and strange procedure (e.g. TRACE) may be suspicious.
http_uri	Requested URI	Is able to disclose attacked endpoints.
http_referrer	Referring page	Is able to demonstrate the source of request; assists in the identification of CSRF-like attacks.
http_version	HTTP protocol version	E.g. 1.1, 2.0. It is used to profile the clients.
http_request_body_len	Length of HTTP request body	Detects large uploads.
http_response_body_len	Length of HTTP response	Here, large downloads or exfiltration can be intercepted.
http_status_code	HTTP response code	Scans may be identified with patterns of numbers: 200 404 500, etc.
http_user_agent	Client browser info	Finds source tool/script. Bots can be impersonators of browsers.
http_orig_mime_types	Original MIME types	Signifies the content nature.
http_resp_mime_types	Response MIME types	Explores the types of downloaded files. Risky when performable.
weird_name	Anomaly tag name	Anomaly tags such as bad_TCP_checksum are supposed to be illuminated by Bro/Zeek. Highlights issues.
weird_addl	Additional info on anomaly	A bit more explanation of why it is weird. Aids in the explanation of anomalies.
weird_notice	Should it raise alert?	Boolean; indicates that this anomaly has to be flagged.
label	Ground truth label	0 defines normal 1 defines attack.
type	Attack type or normal	Specific classes like DDoS, Injection, Normal. Used for classification.

Table 3.1: Descriptions of Important Features Used in Intrusion Detection

3.4 Relevance of Features and Outlier Behavior

Mutual information was used in an information-theoretic analysis of the relevance of features with respect to the target variable. Timestamp, source IP bytes, destination IP bytes, and destination port features showed the maximum mutual information values, which means they are highly related to attack types.

Features with a high rate of anomaly were detected using outlier identification via standardized Z-scores. As an example, dst port and src port had a larger proportion of statistical outliers, indicating abnormal usage habits, which could be related to attacks.

3.5 Adversarial Leakage and Feature Elimination

An evaluation of leakage showed a leakage score of 0.700, which is above the widely used cut-off value of 0.55. That meant that certain features were too strongly associated with the target labels, and models may be able to "cheat" by picking up on spurious shortcuts. To reduce this, destination port, duration, connection state, source IP bytes, IP byte ratio features were eliminated. This was necessary so that the models did not merely learn the decision boundaries (and become overfitted) on the artifacts in the data.

3.6 Preprocessing Workflow

In order to make the dataset usable with machine learning models, the following preprocessing steps were used:

- Target Encoding: The class names were changed to a numerical format to be able to classify them.

- Feature Selection: The features that had very low variance were dropped, since they offered minimal or no discriminative ability.

- Dataset Partitioning: The data was divided into training and testing subsets based on a stratified sampling strategy to maintain the class balance in both subsets.

- Normalization: All numeric attributes were brought to the same range within the min-max normalization, which is important to avoid the situation when a specific attribute has an overproportional impact on model training.

- Feature Filtering: As a post-processing step (final manual step) feature filtering was done to eliminate potentially leaky features based on domain knowledge and leakage diagnostics.

Chapter 4

Model Implementation

4.1 Decision Tree Architecture

Decision trees are also referred to as non-parametric and supervised algorithms, and it finds usage in both classification and regression tasks. Such a model has a treelike hierarchical structure that contains a root node, branches, the internal nodes and all the leaf nodes.

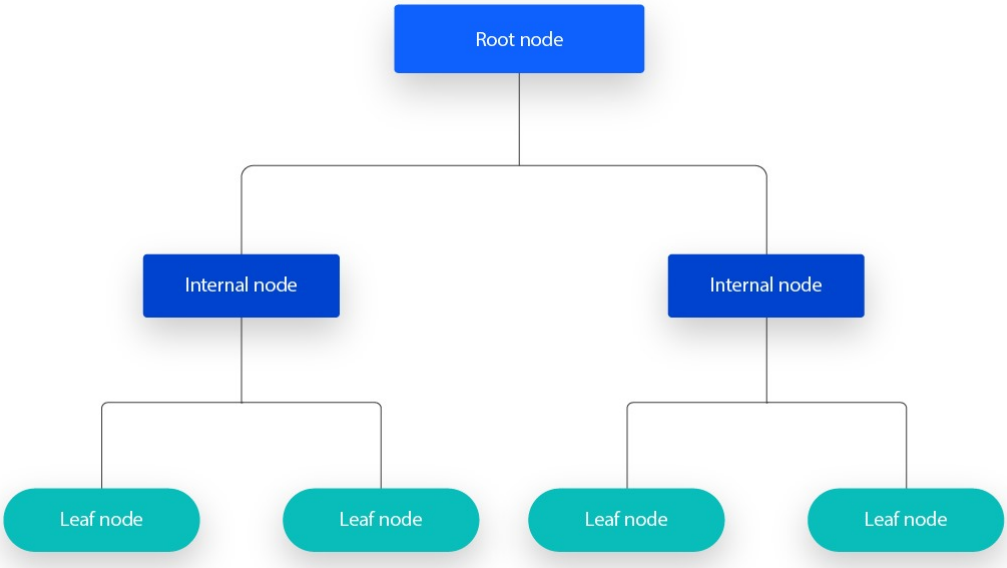

Figure 4.1: Decision Tree Architecture

4.1.1 Decision Tree Drive

The decision tree resorts to a nested algorithm that addresses the dataset as a recursive set of subgroups. As a rule at best split ever there are child nodes that each ideally contain data from only 1 class (for classification). The tree examines each of its features and decides which feature and which threshold value will yield the maximum information gain given by either Gini Impurity (default) or Entropy.

4.1.2 Decision Tree Structure

Nodes: As all the fundamental components of a tree, nodes are used for making a decision to split the dataset based on some feature.

Branches: The links from a parent node to child nodes are the results of decision-making, like: if feature A is greater than a given valu**e o left**, otherwise go to right.

Depth: The degree of a tree is the amount of levels it has among the root node and the furthest most successive leaf node.

Pruning: Preventing the tree from overfitting to the training set by restricting its depth or minima parent-child connections.

4.1.3 Decision Tree Model Formula

Gini Impurity: It is asserted that the Gini impurity formulation is applied in obtaining a measure of the level of the splits in the DecisionTreeClassifier. For Gini Impurity, the formula is expressed in terms of where pi is the probability of a certain instance class in a particular node.

$$\text{Gini Impurity} = 1 - \sum_{i} (p_i)^2$$

4.1.4 Decision Tree Model Related Paper Review

According to the International Journal of Computer Sciences and Engineering (2018), A tree, consists of a root, branches, and leaves. This same pattern is observed in the Decision Tree. It has root nodes, branches and leaf nodes. For every internal node, testing an attribute is done, the result of that test is found in a branch and the explanation of the outcome rests within the leaf node. A root node refers, as the name suggests, the top most node in a Tree and a node to the other nodes. A decision tree can be defined as a tree which gives decision-making of one attribute at each of its nodes (which displays a feature), with each of its links being a rule, and with each of its leaves being an outcome. Because decision trees are designed to mimic human-level thinking they are very easy to use, most people can pick up the data and be able to make good interpretations of the data. So the whole idea is to build a tree structure such as this for the entire dataset and in turn produce a single outcome for every leaf node [5].

4.2 Random Forest Architecture

Building and training the various individual decision trees is done using an ensemble learning approach called Random Forest which acts together as a more powerful model and reduces the chances of overfitting. From the architecture, the dataset comprises two sets consisting of a training set and a test set where many bootstrapped subsets are used for training individual decision trees. Each tree makes its own prediction and all the predictions are averaged out for regression or a majority response is sought in case of classification for the final output. This method takes advantage of the diversity of the trees and hence a stronger and more diverse model is achieved.

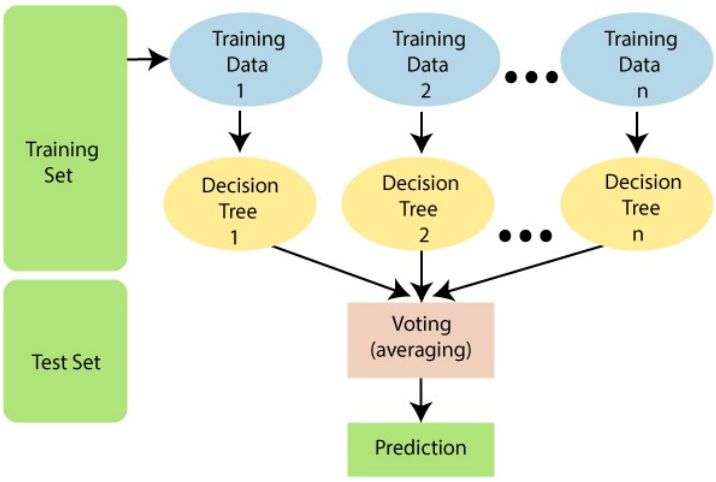

Figure 4.2: Random Forest Architecture

4.2.1 Random Forest Drive

Random forest algorithms are considered to be the most supervised ML algorithms, as it enhances the predictive precision while avoiding the risk of overfitting by ensembling numerous decision trees. The way this algorithm works is firstly by building many decision trees from different random training samples of the data, thus promoting diversity among the trees. During the process of training any particular tree, a unique subset of features is used to make sure no one tree's predictions dominate the model. For prediction, the results of all the trees are combined, and the final result is reached either by majority rule for classification problems or by averaging for regression's models' predictions. This integrated strategy allows for complex data relationships to be learned in the model, and Random Forests are well suited for many tasks such as those found in finance or health care while still being able to perform with certain levels of interpretability with the help of feature importance values.

4.2.2 Random Forest Structure

Collection of data: Assemble and prepare the dataset while constraining the missing value and categorical variables.

Initialization of models: Based on the featured and random sampling of the training dataset, several decision trees are produced.

Training of models: Each tree is trained independently on a certain sample space which has a tree of its own.

Prediction: In common for a class of trees, each tree casts its vote and the majority's decision prevails. In common, in regression for prediction, the arithmetic mean of the all trees is returned.

Evaluation of model performance: Examine the assessment figures of the model and compare them with other evaluation metrics such as accuracy, precision, recall, and F1 score.

Model-Variable importance: It can also be utilized in estimating the importance of different variables, when making the prediction.

4.2.3 Random Forest Model Formula

The accuracy formula evaluates the performance of a classification model when aimed at a specific task, let's say predicting correct outcomes with Random Forest. In this case, when we refer to the Random Forest classifier, this is a formula that assesses how many of the predictions made by an aggregate of decision trees are correct.

The accuracy of a model is given by:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Predictions}} = \frac{TP + TN}{TP + TN + FP + FN}$$

Where:

- TP = True Positives: Instances correctly predicted as the positive class.
- TN = True Negatives: Instances correctly predicted as the negative class.
- FP = False Positives: Instances incorrectly predicted as the positive class.
- FN = False Negatives: Instances incorrectly predicted as the negative class.

The formula indicates the practical applicability and accuracy of the Random Forest models' predictions in respect of the features of the classification objects.. Although accuracy is a useful metric, additional metrics such as precision and recall are certainly necessary for better evaluation when the class distribution is skewed.

4.2.4 Random Forest Model Related Paper Review

In this article, Khammas(2020) propose in his paper titled "Ransomware Detection using Random Forest Technique" implements a static analysis method for detecting ransomware and gives a new paradigm. This technique uses frequent pattern mining and it is very simple and does not take a lot of time to perform like dynamic analysis. Functionas are taken straight out of executed documents raw byters by this method. Feature selection is carried out using the Gain Ratio technique while screening classifying is done using the Random Forest classifier. The paper focuses on how different numbers of trees and seeds affect the classifier's performance with the conclusion that a hundred trees and one as a seed performs the best. The presented model is accurate by an impressive 97.74 percent with a 1.37 second processing time. This is very helpful as it eliminates the tedious disassembling stage in opcode based detection techniques that in turn gives quicker results at high accuracy levels. The accuracy stated above in regard to the paper means the Random Forest classifier made valid predictions with an accuracy level of approximately 98 out of every 100. Given the level of accuracy the model possesses it is apparent that the model can easily distinguish between ransomware and benign files which indicates that this model can be useful for cybersecurity. Random Forest is compared to other classifiers namely AdaBoost, and Bagging in the paper where it does not fail to outperform its competitors every time clearly [6].

4.3 Naive Bayes Architecture

The Naive Bayes classifier is an uncomplicated probabilistic classifier based on Bayes Theorem with consideration of the independence of all the features used in the input. This model can easily be applied to a variety of classifications as it doesn't fit a complex structure.

Input Layer: Each input data in the dataset such as latitude, longitude e.t.c is an input.

Hidden Layer: Naive Bayes does not possess any layers other than the feature, it simply uses different probability rules to the features.

Output Layer: Determining output class prediction such as normal or anomaly in color on the basis of the label is the function of the output layer.

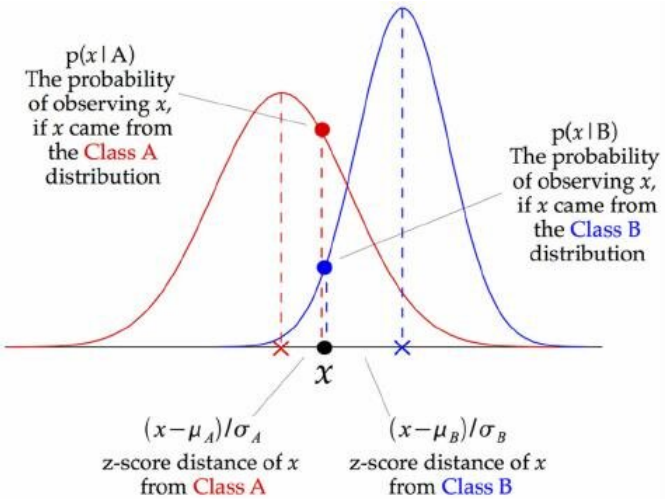

Figure 4.3: Naive Bayes Architecture

4.3.1 Naive Bayes Model Drive

Bayes' Theorem gives a mathematical rule that works for every type of classification, as well as the Naive Bayes Classifier which resulted from it. What distinguishes this theorem, among other known ones, is that this one works more like looking at features rather than classes i.e. given a certain class, what are its features? What are the features that allow inference to be made to that particular class? This is the basic definition of the Naive Bayes Classifier, which can thus be easily applied to multi-class problems. The reason it is called "Naive" is because it makes a very strong assumption that all features are independent. Because this makes the model much simpler it also means that there may be some inaccuracy in some cases. Simply put, the model computes the chances of each feature occurring within a particular class, and multiplies those chances resulting in the probability with the highest score being regarded as the likelihood of it being that class.

4.3.2 Naive Bayes Model Structure

Input layer: The features mean the independent variables in the dataset.

Model Computation: For each input feature, the model computes probabilities using conditional probabilities and multiplies them together with the prior probability of the class.

Output layer: This produces the probability of each class label.

Steps in the Naive Bayes model structure

Training phase: The model learns probabilities from training data, including likelihood and prior probabilities.

Prediction phase: The model determines the probability of each class given the input Prediction Phase. The model predicts input features and uses the features corresponding to the predicted class with the maximum probability for its prediction.

Types of Naive Bayes Classifiers: Multinomial Naive Bayes: This is used when the categorical or discrete data are required.

Bernoulli Naive Bayes: Designed for use with binary data consisting of 0 and 1.

Gaussian Naive Bayes: This model assumes normal distribution with continuous data.

4.3.3 Naive Bayes' Theorem Model Formula

The formula for Naive Bayes is based on Bayes' Theorem.

The Bayes' Theorem:

$$P(\frac{Y}{X}) = \frac{P(X|Y)P(Y)}{P(X)}$$

The Naive Bayes Theorem:

$$P(\frac{C_K}{X}) = \frac{P(X|C_K)P(C_K)}{P(X)}$$

Where:

$P(\frac{C_K}{X})$ is the posterior probability of class (C_K) given feature vector (X).

P(X | (C_K) is the likelihood of the feature set(X) given class (C_K).

P(C_K) is the prior probability of class (C_K).

P(X) is the prior probability of the feature set (this is typically ignored as it doesn't affect the class comparison).

The assumption is that each feature is independent, so for feature set X = $\{x_1, x_2, \ldots, x_n\}$

$$P(\frac{C_K}{X}) = P(C_K) \prod_{i=1}^{n} P(x_i|C_K)$$

4.3.4 Naive Bayes' Related Paper Review

"A New Two-Phase Intrusion Detection System with Naïve Bayes Machine" Authors: Monika Vishwakarma and Nishtha Kesswani

This perspective paper gives a new take on the two-phase Intrusion Detection System (IDS) targeting optimal security for every Internet of Things (IoT) device. It takes care of class imbalance and detectability for various datasets. This new approach seeks to refine Naive Bayes based Intrusion Detection systems to suit IoTs that are resource constrained. The first phase of this classification process takes data as nominal, binary, integer or float and categorizes them using a Naïve Bayes classifier and a majority vote. The later stage deals with deeper inspection of the normal classified objects using an unsupervised elliptic envelope. Results of testing on NSL-KDD, UNSW-NB15 and CIC-IDS2017 datasets showed the system produces results accuracy of 97 percent, 86.9 percent, and 98.59 percent respectively and it also handles uneven class distributions by using weight initialization methods but find it difficult with multiclass classification [24].

4.4 Gradient Boosting Architecture

Gradient Boosting is a stage-wise algorithm (ensemble method) that builds an additive predictive model through a combination of many weak learners (most often decision trees). It has sequential architecture: at each step a new tree is trained to reduce the errors of the collective ensemble to date. It does this by performing gradient descent in the space of functions, the aim being to minimize a loss function by fitting models to the residuals of previous iterations. This forward, iterative architecture enables the model to iteratively refine the predictions bias and variance.

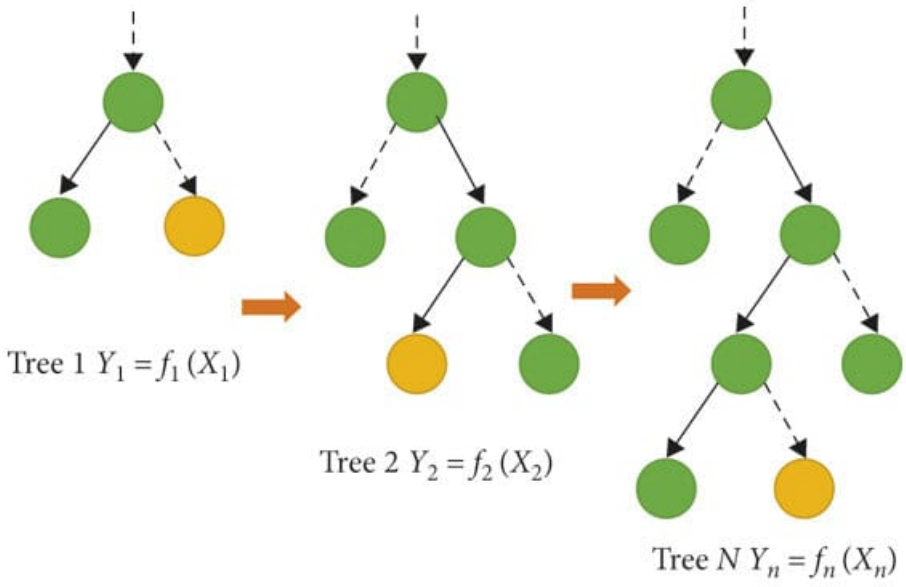

Figure 4.4: Sequential Tree Learning in Gradient Boosting

4.4.1 Gradient Boosting Drive

Gradient Boosting is motivated by the desire to enhance the performance of simple learners via a corrective sequence combination. In contrast to parallel ensemble techniques such as bagging, boosting is a method which aims at improving errors, which in many cases results in better accuracy. Nevertheless, the original Gradient Boosting has had its recognized weaknesses in the form of being sensitive to noise, requiring more time to train, and having a propensity to overfit unless care is taken to apply regularization methods.

4.4.2 Gradient Boosting Model Formula

Gradient Boosting ensemble prediction takes the form of:

$$\frac{\partial}{\partial \gamma} \sum_{x_i \in R_{jm}} (y_i - F_{m-1}(x_i) - \gamma)^2 = 0$$

$$-2 \sum_{x_i \in R_{jm}} (y_i - F_{m-1}(x_i) - \gamma) = 0$$

$$n_j \gamma = \sum_{x_i \in R_{jm}} (y_i - F_{m-1}(x_i))$$

$$\gamma = \frac{1}{n_j} \sum_{x_j \in R_{jm}} r_{jm}$$

The negative gradient of the loss function is trained on each new model hth-t so that the ensemble is converted iteratively towards the optimum function.

4.4.3 Gradient Boosting Model Related Paper Review

The formalization of Gradient Boosting is owed to the additive logistic regression seminal paper by Friedman (2001). Ensemble learning has been since based on that model. Though highly effective, its disadvantages of scale and control overfitting led to the development of more recent frameworks like XGBoost and Light GBM which further expounded on its core concepts but with better computational and statistical tricks.

4.5 XGBoost Architecture

XGBoost (an abbreviation of extreme gradient boosting) is a greatly optimized gradient boosting. It contains a number of architectural improvements: regularized objective function, second-order gradient optimization, parallelized tree building and sparsity-aware data management. XGBoost builds decision trees leaf-by-leaf, like in classical boosting, but with better memory usage, computational efficiency

and numerical stability.

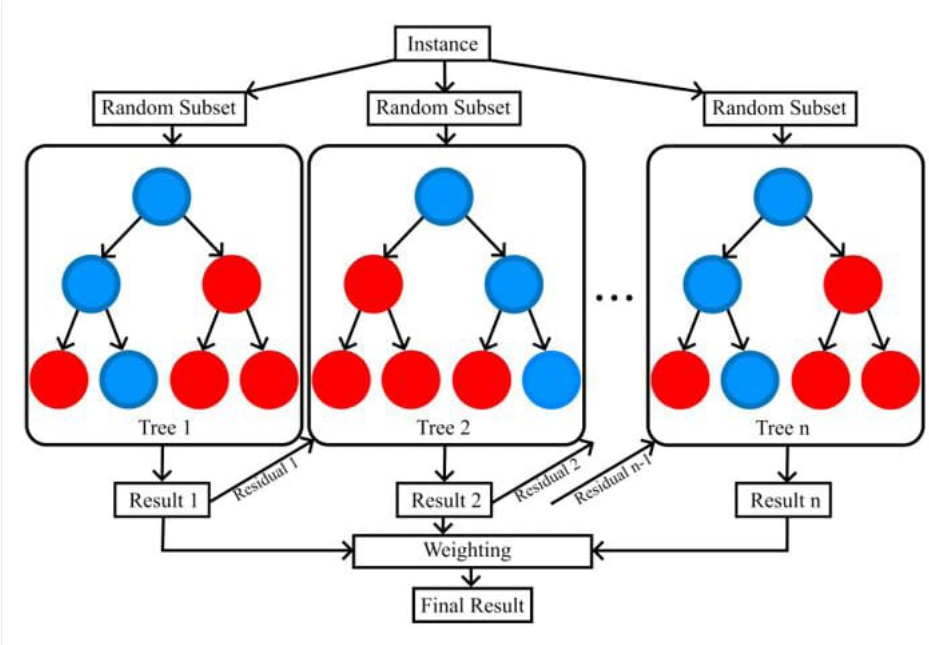

Figure 4.5: Diagram of XGBoost Ensemble Learning Framework

4.5.1 XGBoost Drive

XGBoost is driven by the desire to resolve the performance limitations of the conventional gradient boosting. It was developed to have high scalability, avoid overfitting due to L1 and L2 regularization and offer more control over the complexity of trees. Its resilience and stability in structured data application have predisposed it to be one of the most embraced models in data science and industry.

4.5.2 XGBoost Model Formula

XGBoost training goal is set as:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i)$$

Where:

- $\hat{y}_i$ is the final predicted value for the i^{th} data point
- K is the number of trees in the ensemble
- $f_k(x_i)$ represents the prediction of the k^{th} tree for the i^{th} data point

Such a formulation is a compromise between predictive accuracy and model complexity that helps to reduce overfitting and increases generalization.

4.5.3 XGBoost Model Related Paper Review

The XGBoost was introduced by Chen and Guestrin (2016), who paid attention to the optimization of the system and the algorithm. The paper introduced columns block compression technique, cache awareness technique and sparsity handling which allowed XGBoost to become superior to other models in large machine learning competitions. The fact that it can combine accuracy, regularization and scalability properties led it to be very much applicable to the intrusion detection use case described in this paper[2].

4.6 LightGBM Architecture

Light Gradient Boosting Machine (LightGBM) is a framework (boosting) created by Microsoft, with a focus on efficiency and speed. As opposed to level-wise growth in conventional boosting, LightGBM uses leaf-wise tree expansion strategy, where the leaf with the largest information gain is expanded. It also takes on histogram based feature binning that speeds up training and lowers memory usage. Such an architectural design makes LightGBM scale to very large dataset with competitive performance.

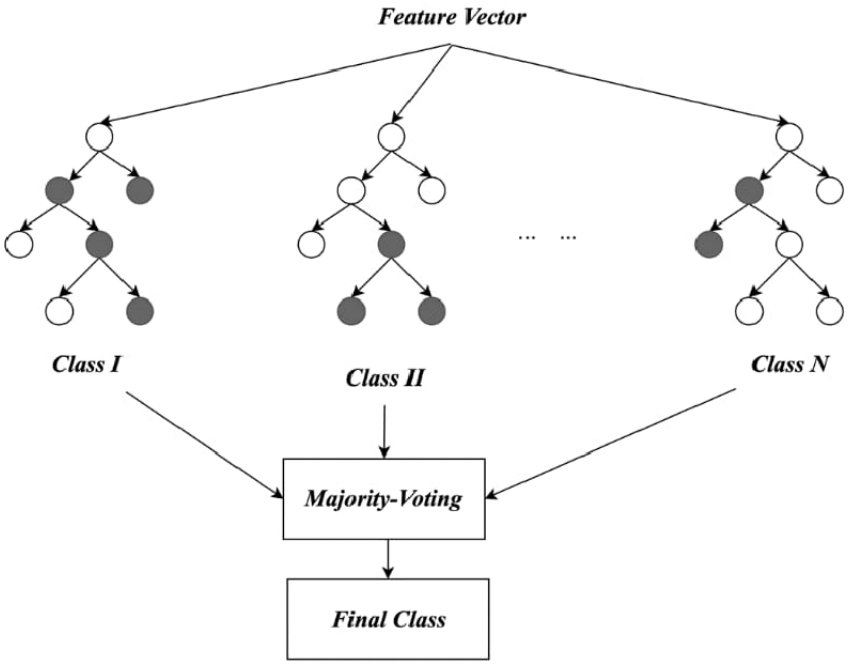

Figure 4.6: LightGBM Classification Architecture

4.6.1 LightGBM Drive

LightGBM is an enhancement of XGBoost and gradient boosting in general, which faces problems of scalability. It is designed to take large-scale high-dimensional dataset into consideration and can train faster and with less memory usage. It has native support of GPU training, distributed computing, and categorical feature processing, which is why it is best suited to industrial-grade applications like network intrusion detection.

4.6.2 LightGBM Model Formula

The LightGBM objective is similar to second-order Taylor expansion as XGBoost:

$$\mathcal{L} = \sum_{i=1}^{n} \left[l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) \tag{4.1}$$

where:

- $l(y_i, \hat{y}_i^{(t-1)})$ is the loss between the true label y_i and the prediction at iteration $t-1$,
- $f_t(x_i)$ is the prediction of the model at iteration t,
- $g_i = \frac{\partial l(y_i, \hat{y}_i)}{\partial \hat{y}_i}$ is the first derivative (gradient),
- $h_i = \frac{\partial^2 l(y_i, \hat{y}_i)}{\partial \hat{y}_i^2}$ is the second derivative (Hessian),
- $\Omega(f_t)$ is the regularization term for controlling the complexity of f_t.

The optimization employs histogram based algorithms and leaf-wise tree growth.

4.6.3 LightGBM Model Related Paper Review

Ke et al. (2017) proposed LightGBM, which is an extremely efficient alternative of XGBoost. The authors conducted comparative experiments which showed that LightGBM could training time up to 20 times faster on large scale datasets and that it also offered similar or better accuracy. Due to its scalability and speed, it is especially beneficial in time-sensitive security systems such as intrusion detection system (IDS), where accuracy and speed of inference are both important[4].

Chapter 5

Model Evaluation

5.1 Overview of Evaluated Models

In the present work, a wide range of machine learning classifiers have been used to determine their effectiveness in identifying different forms of network intrusions. The models consisted of both conventional algorithms and the modern ensemble-based learners. The main aim was to find out classifiers that not only have high predictive performance but also have good generalization capability across different kinds of intrusions.
The models taken into consideration were:

- Traditional classifiers: Logistic Regression, Support Vector Machine (SVM), Naive Bayes, K-Nearest Neighbors (KNN), Decision Tree.
- Ensemble classifiers: Random Forest, Gradient Boosting
- Boosted learners: XGBoost and LightGBM
- Hybrid ensembles: All sorts of combinations of XGBoost, LightGBM, Random Forest, and Gradient Boosting with soft-voting logic

The models were evaluated using several performance measures: accuracy, F1-score, and ROC AUC, and a multiclass classification approach was aimed at six classes of traffic.

5.2 Best Performing Base Models

Of all standalone models, three have received much better results:

- As an individual model, XGBoost got the highest results with an accuracy of 97.20 percent, an F1-score of 0.9721, and a ROC AUC of 0.9983. It owes its good performance to the fact that it has good regularization mechanisms and

a tree-based architecture that deals with non-linearity and feature interactions.

- Close behind was LightGBM with 97.18 percent accuracy and almost the same ROC AUC of 0.9983. It displayed good generalization and minimal training time as a gradient-boosting framework that is optimized to be fast and efficient.

- Random Forest also showed good results, 97.15 percent accuracy and ROC AUC 0.9960. It provided consistent performance among the classes but was slightly outperformed by boosting models in recalling some categories of attacks, like scanning and XSS.

Such models proved especially successful because they can represent arbitrarily complicated decision boundaries, and because they are not sensitive to feature noise or outliers.

5.3 Traditional Models performance

The performance of traditional models was quite low:

- Logistic Regression (accuracy=57.41 percent) was not able to handle the nonlinear separability of classes.

- Naive Bayes and SVM were around 52 59 percent accuracy and they were not effective in dealing with feature interactions and high dimensions of categorical variables.

- Although KNN and Decision Tree had improved over the linear models, they could not match the ensemble methods especially on generalization within the minority classes.

These findings support the significance of ensemble as well as boosting techniques in high-complexity areas like network intrusion detection.

5.4 Hybrid Ensemble Models

In order to achieve even better results of the classification, a number of hybrid ensemble models were formed. These models combined the merits of more than one high-scoring classifier through a soft voting scheme. The important hybrid combinations and their results are as follows:

- **Hybrid 3 (XGBoost + LightGBM + Random Forest)**
Overall performance of this model was the best with an accuracy of 97.25 percent, an F1-score of 0.9726 and an ROC AUC of 0.9984. The ensemble of the varied base learners helped in the minimization of overfitting along with producing better robustness. It gave importance to both the statistical and structural learning orientations and performed well when class boundaries were complicated.

- **Hybrid 2 (XGBoost + LightGBM)**
This architecture provided a good compromise between performance and computing efficiency. It obtained an accuracy of 97.23 percent almost the same as Hybrid 3 but with simpler models. It is simple hence can be used in real time or resource restricted systems.

- **Hybrid 4 (XGBoost + LightGBM + Random Forest + Gradient Boosting)**
Although this ensemble is the most complex one, the model did not show significantly better results compared to the others. It obtained an accuracy of 97.22 percent which indicated that the extra model (Gradient Boosting) introduced redundancy instead of additive value. Its slower inference time can cancel out its accuracy marginalims.

On the whole, these hybrid models showed that model diversity helps generalization, particularly in multiclass when there are delicate differences between malicious behaviors.

5.5 Top Three Models Comparison

Hybrid 3 (XGBoost + LightGBM + Random Forest), Hybrid 2 (XGBoost + LightGBM), and XGBoost (Base) were the best models among all models tested. Although their evaluation measures were very similar, they both had very different strong points and trade-offs with regard to their architecture, complexity, and applicability.

The best overall performance was recorded by Hybrid 3 with an accuracy of 97.25 percent , an F1-score of 0.9726, and a ROC AUC of 0.9984. What made it successful is that it generated a diverse ensemble, it launched a combination of the gradient-boosting power of XGBoost and LightGBM with the variance-reduction of Random Forest. Such diversity has allowed this model to learn both structured and unstructured decision boundaries, resulting in a model that is strong, especially in classifying edge cases like scanning and injection attacks. The ensemble was also found to be consistent across all classes, thereby making it a very reliable solution for intrusion detection systems.

Hybrid 2 was a little slower in performance, but had an attractive trade-off be-

tween accuracy and efficiency. Nevertheless, having the same ROC AUC as Hybrid 3 but with an accuracy of 97.23 percent, it kept most of the predictive power but decreased the computational burden. Restricting the ensemble to only XGBoost and LightGBM, which are highly optimized gradient boosting frameworks, made the model quicker to train and less resource hungry during inference time. This enables Hybrid 2 to be more realistic to be use in those occasions where computation facilities or reaction time are a constraint, as in the case of real-time network monitoring systems.

As a single classifier, XGBoost got an accuracy of 97.20 percent and a ROC AUC of 0.9983, which is the highest amongst the single models used in the experiment. Its regularized gradient-boosting methodology enabled it to generalize well on all the categories of attacks (both frequent and rare classes). Besides, XGBoost has a reputation for being resistant to missing values, overfitting, and being interpretable with the help of such tools as LIME, which makes it an outstanding baseline to compare to and construct ensembles.

As a conclusion, Hybrid 3 should be used when the primary goal is performance, and computing resources are not a problem. Hybrid 2 offers the best compromise between resource usage and latency-critical applications without sacrificing much accuracy. In the meantime, XGBoost can be praised as a powerful, explainable single-model algorithm that serves as a good starting point both in a standalone application and as a basis for an additional ensemble.

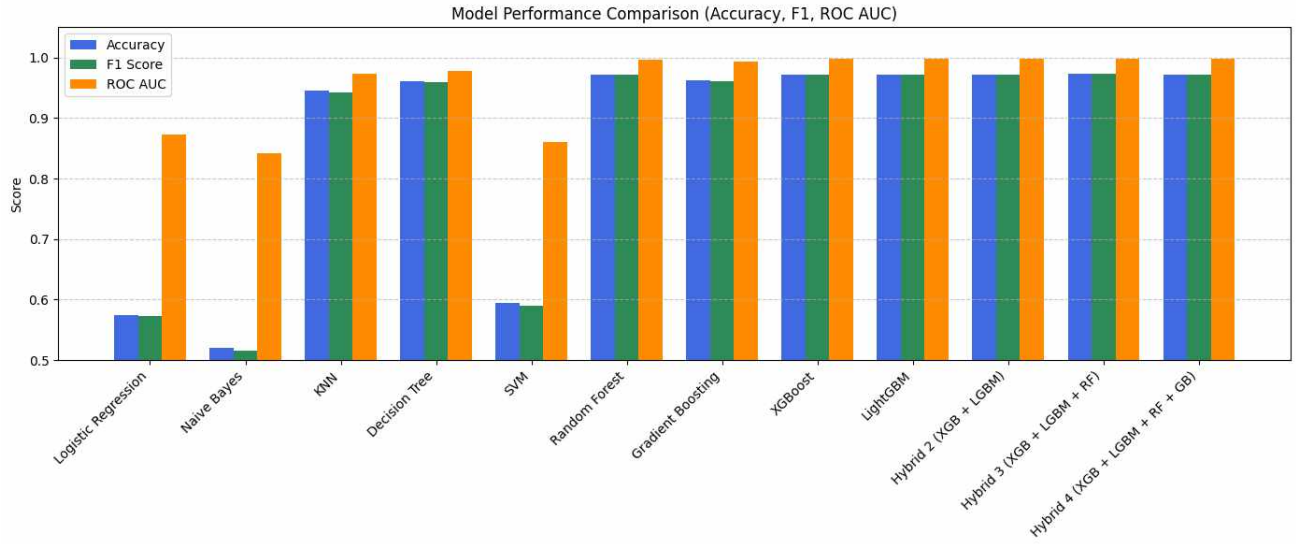

Figure 5.1: Model Performance Comparison (Accuracy, F1, ROC AUC)

5.6 Major Observations

- Techniques based on boosting (XGBoost and LightGBM) were always at the top of the results in comparison to other models, whether as single models or as elements of hybrid ensembles.

- Hybrid models resulted in small, yet significant increases in performance measures, especially in the classification of edge cases and minority types of attacks.

- More models did not necessarily mean better performance in an ensemble, indicating the value of a balanced ensemble design rather than simplicity in an ensemble.

- The old models did not work well on this front, implying that the linear and probabilistic assumptions do not work well with the contemporary network intrusion data, which have nonlinear and high-dimensional representations.

Chapter 6

LIME Architecture and Explainability Analysis

6.1 LIME Architecture

LIME (Local Interpretable Model-Agnostic Explanations) is a post-hoc interpretability framework, which can be used to explain the predictions of any black-box model in a human-interpretable format. Architecturally, LIME fit a local approximation model (typically a simple linear regression) around individual prediction instance. It achieves this by adding noise to the input sample to create artificial neighbors and asking the complicated model to predict on the neighbors. These perturbed samples are then used to fit the interpretable model with proximity weights, providing explanation of the original model reasoning on instance-by-instance level.

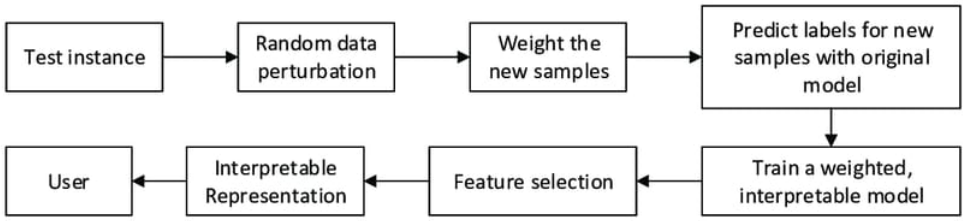

Figure 6.1: Workflow of the LIME Framework

6.1.1 LIME Drive

The major driving force behind LIME is the need to balance the gap between model performance and trust. Although the state-of-the-art models, such as XGBoost and LightGBM achieve high accuracy, their internal working is not easily interpretable. This black box characteristic is a shortcoming when it comes to high stakes areas such as cybersecurity as stakeholders require explanations and justification of automated decisions. LIME can be used to manually examine each prediction on a granular level and offers confidence and transparency without altering the original

model.

6.1.2 LIME Explanation Formula

Each of the following objectives is used by LIME to optimize a local surrogate model:

$$E(x) = L(f, g, \Pi_x) + \Omega(g) \tag{6.1}$$

where:

- f is the original black-box model,
- g is the interpretable surrogate model,
- Π_x defines the locality around the instance x (e.g., using a kernel function),
- $L(f, g, \Pi_x)$ measures how unfaithful g is in approximating f near x,
- $\Omega(g)$ is the complexity penalty ensuring that g remains interpretable.

This form guarantees that the explanation model remains loyal to the original model in a small vicinity around the input case.

6.1.3 LIME Model Related Paper Review

LIME was proposed in the popular paper of Ribeiro et al. called "Why Should I Trust You?" in 2016. The research discussed the importance of model transparency and showed that LIME could be successfully applied to various types of data: texts, images, and tabular data. Since then, LIME has formed a foundation of Explainable AI (XAI), which provides local, instance-specific explanations which can be used by practitioners to validate, debug and deploy machine learning models in sensitive areas. In this thesis, LIME was used on top of the well-performing models such as XGBoost and Hybrid ensembles to produce instance-wise feature attributions that enhanced interpretability and accountability of the IDS framework [3].

6.2 LIME Overview

The field of cybersecurity and especially network intrusion detection systems (IDS) accuracy is not the only measure that needs to be taken. Model interpretability is also of critical importance when models are deployed in a sensitive area like critical infrastructure, enterprise firewalls, or real-time monitoring systems. To solve this, the paper included LIME (Local Interpretable Model-Agnostic Explanations) to explain the predictions of the best models.

LIME works on building a local surrogate model approximately around a prediction. It perturbs (or shakes) the input data and notes the impact on the prediction outcome, hence assigning importance scores to individual features. The method enables researchers and practitioners to see which features had the greatest influence

on a particular classification, and in which direction (positive or negative).

In this research, LIME is explained in the following models:

- XGBoost (Base Model)
- Hybrid 2: XGBoost + LightGBM
- Hybrid 3: XGBoost + LightGBM + Random Forest
- Hybrid 4: XGBoost + LightGBM + Random Forest + Gradient Boosting

None of the traditional models (e.g. Logistic Regression, SVM, Naive Bayes) or standalone ensemble learners (e.g. Random Forest, LightGBM) had any LIME evaluations done, and therefore, the interpretability comparisons are limited to the above models.

6.3 LIME Explanations on XGBoost (Base Model)

The LIME explanation on XGBoost demonstrated heavy dependence on application-layer and statistical traffic characteristics. In the considered cases, the next characteristics were always emerging as important:

- packets-ratio
- http-response-body-len
- http-user-agent
- src-pkts
- src-port
- http-status-code
- http-uri

As an illustration, the values of packets-ratio and http-response-body-len were frequently low and corresponded to anomalous traffic patterns that are characteristics of scanning or injection attacks. Similarly, http-user-agent and http-uri played an important role in detecting XSS-related traffic. XGBoost decision-making process was thus explainable and aligning well with the known attack behavior, making it not just a high-performance model but also a transparent one.

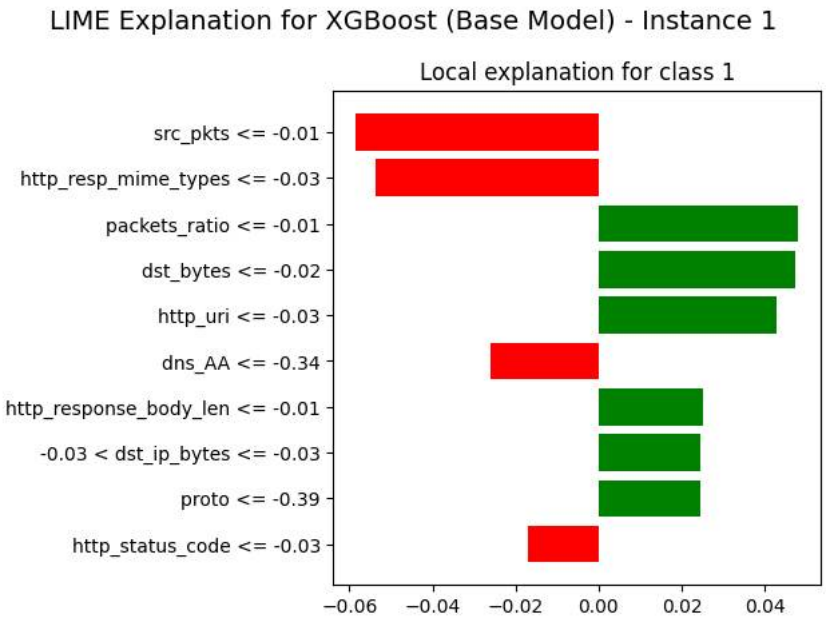

Figure 6.2: LIME Explanation for XGBoost (Base Model) - Instance 1

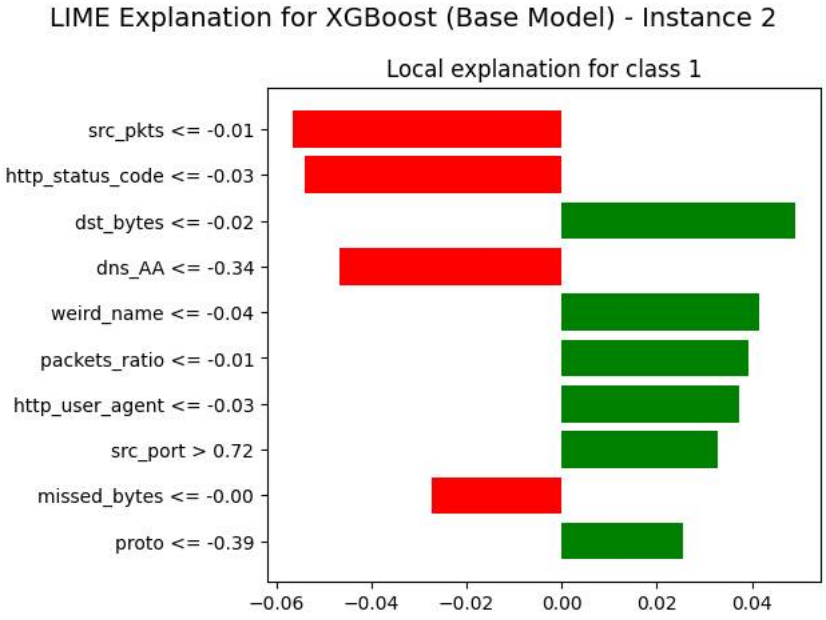

Figure 6.3: LIME Explanation for XGBoost (Base Model) - Instance 2

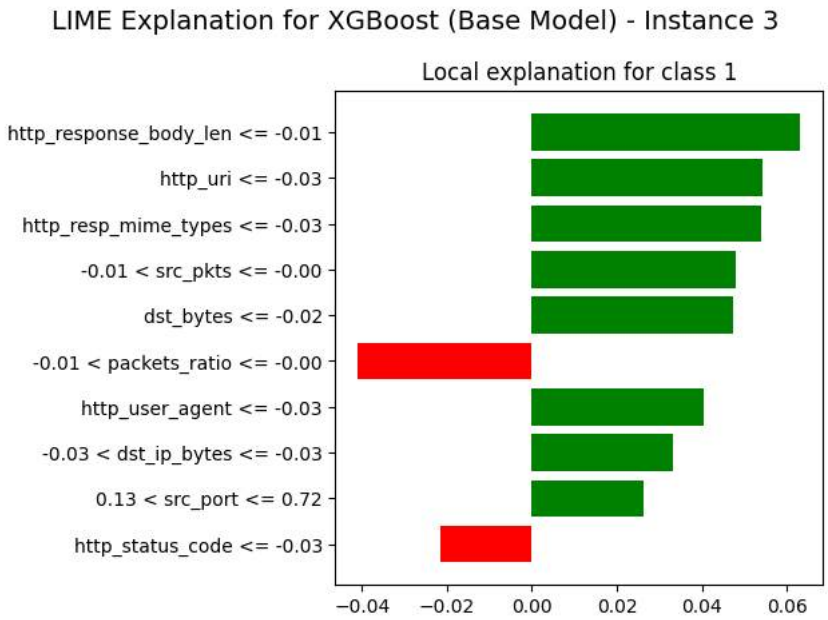

Figure 6.4: LIME Explanation for XGBoost (Base Model) - Instance 3

6.4 LIME Explanation on Hybrid 2 (XGBoost + LightGBM)

The Hybrid 2 model showed somewhat different contributions of features to XGBoost with more focus on several additional protocol-level fields, which can probably be explained by the leaf-wise optimization of trees in LightGBM. The influential features were:

- packets-ratio
- http-response-body-len
- src-pkts
- src-port
- http-user-agent
- http-status-code
- dns-AA
- proto

LIME explations revealed that certain statistical scores (e.g., packet distributions) and categorical features (e.g., DNS flags and protocol identifiers) were the cause of driving classification results. This ensemble model showed more subtle decision boundaries and worked with subtle class differences, such as those between injection and XSS, than XGBoost on its own. The collision of LightGBM added complimentary patterns that diversified the feature importance landscape.

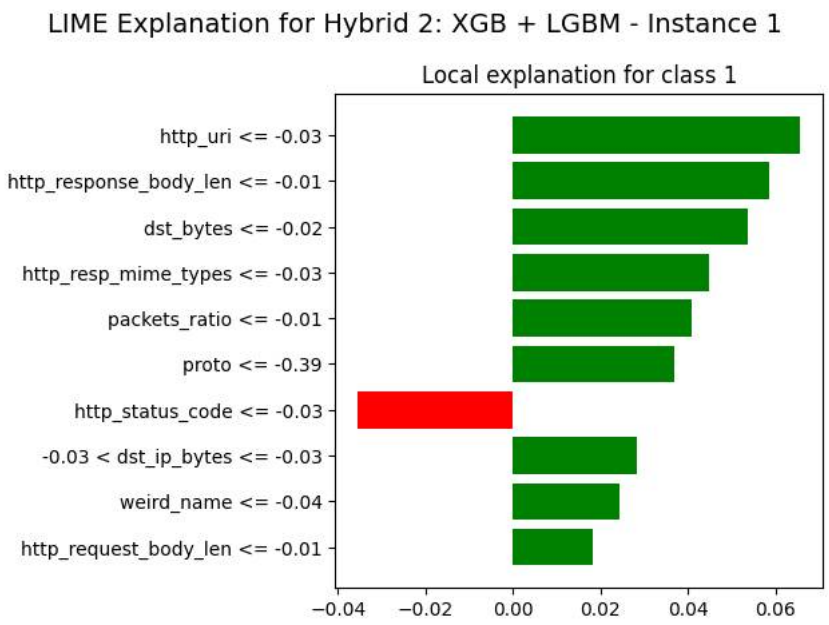

Figure 6.5: LIME Explanation for Hybrid 2: XGB + LGBM - Instance 1

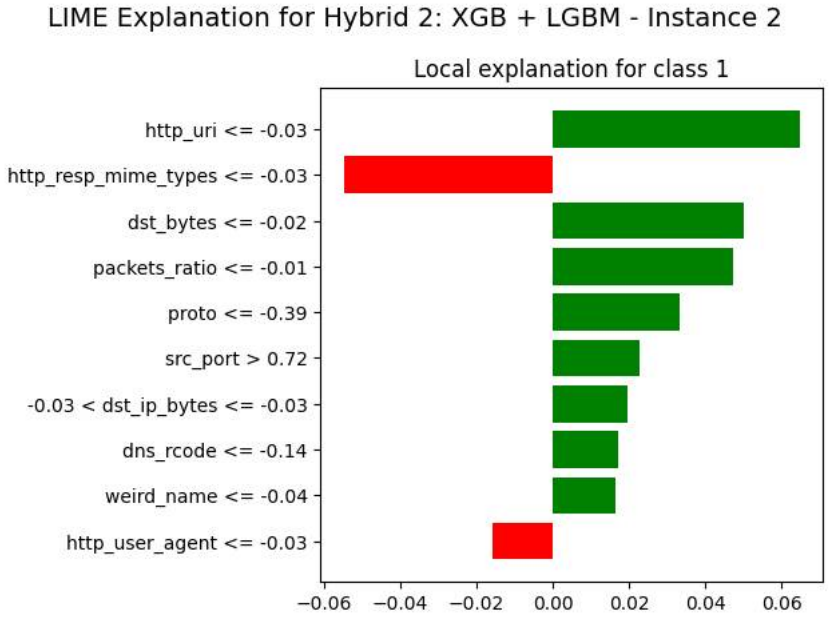

Figure 6.6: LIME Explanation for Hybrid 2: XGB + LGBM - Instance 2

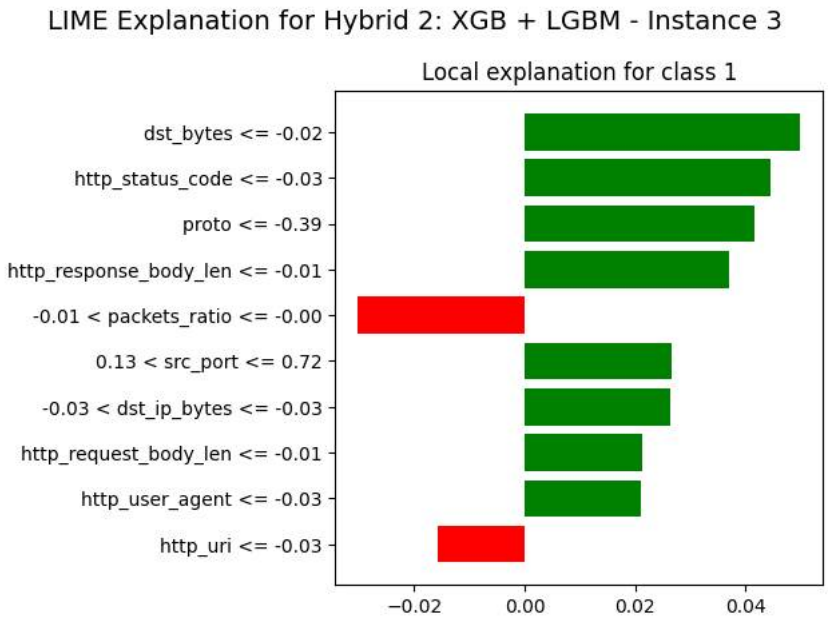

Figure 6.7: LIME Explanation for Hybrid 2: XGB + LGBM - Instance 3

6.5 LIME Explanations on Hybrid 3 (XGBoost + LightGBM + Random Forest)

Hybrid 3 had the most stable and wide range of influential features during LIME evaluations. The ensemble seemed to better balance statistically-related, structural and categorical features, and the following variables commonly appeared in descriptions:

- http-request-body-len
- packets-ratio
- src-pkts
- service
- http-response-body-len
- proto
- dns-rcode

Among others, it was noted that the model was sensitive to an interaction between HTTP payload size and DNS response behavior. This implied that Hybrid 3 was capable of detecting multi-vector attacks, which cut-across network layers. Random Forest introduced stability and reduced the over-reliance on one attribute that resulted in more robust and generalizable explanations.LIME output also indicated that this model was able to effectively capture higher-order interactions between traffic volume and service features, including the ability to differentiate between legitimate high-traffic sessions and denial-of-service activity.

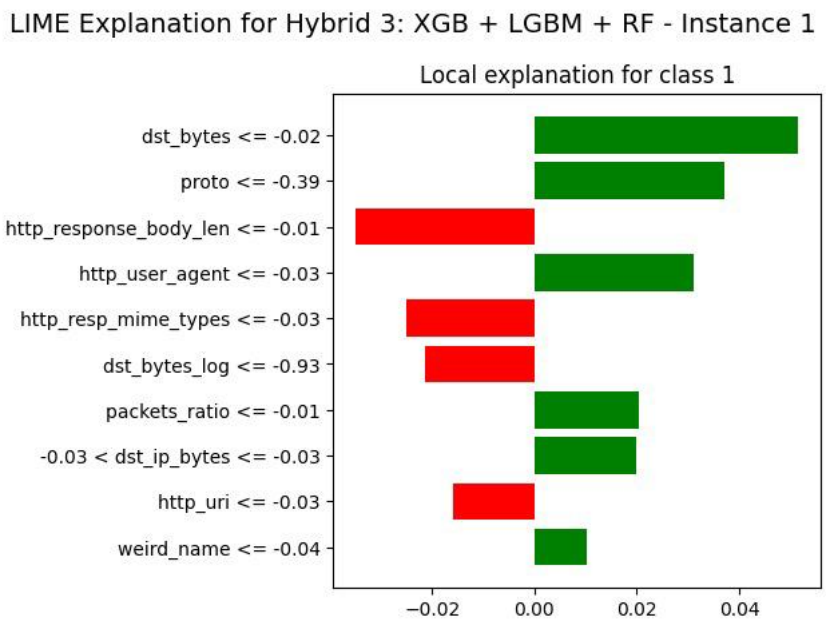

Figure 6.8: LIME Explanation for Hybrid 3: XGB + LGBM + RF - Instance 1

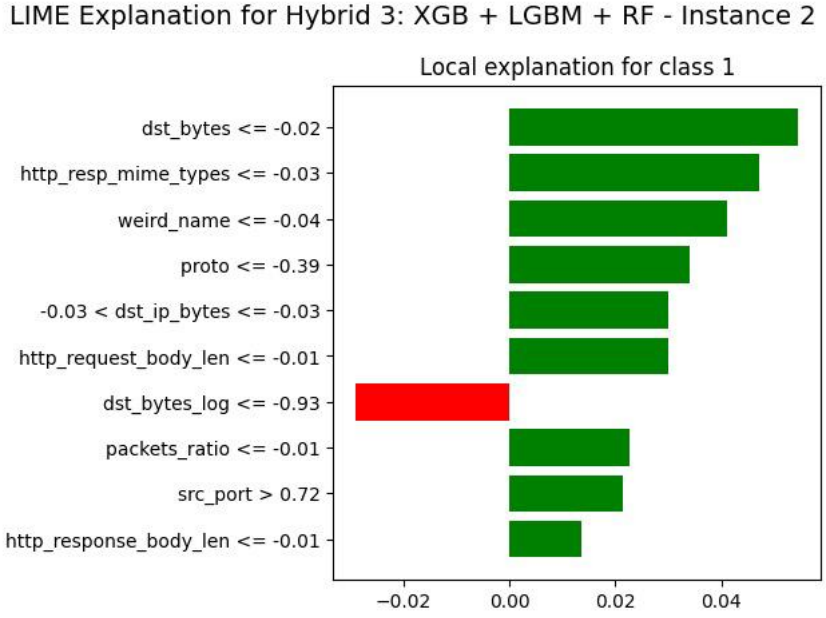

Figure 6.9: LIME Explanation for Hybrid 3: XGB + LGBM + RF - Instance 2

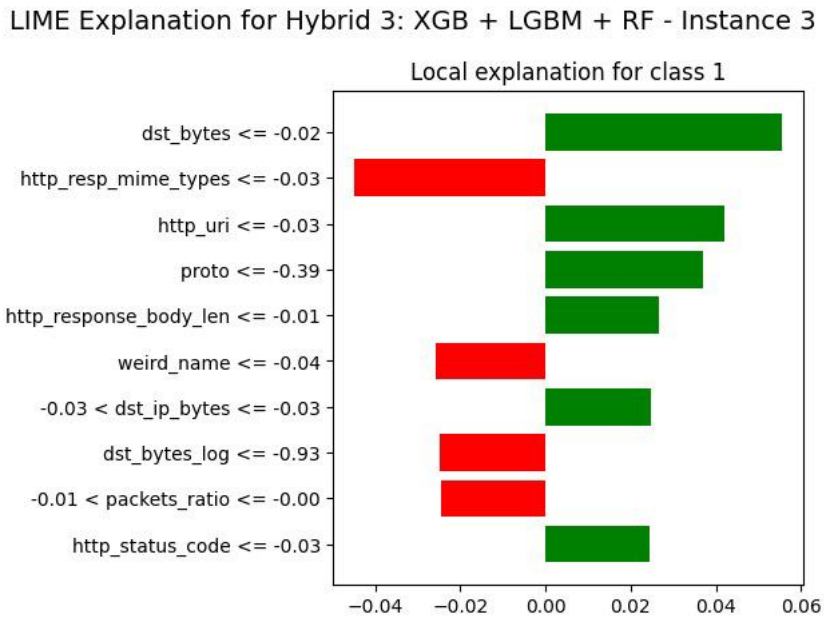

Figure 6.10: LIME Explanation for Hybrid 3: XGB + LGBM + RF - Instance 3

6.6 LIME Explaination on Hybrid 4 (XGBoost + LightGBM + Random Forest + Gradient Boosting)

The Hybrid 4 ensemble, the most complex, exhibited the same patterns of interpretability as Hybrid 3, with minor artefacts of feature overlaps and noise in the LIME explanations. Notable features included:

- src-pkts
- src-port
- http-uri
- http-status-code
- http-resp-mime-types
- weird-name
- dns-qtype
- proto

Despite the fact that this model had continued to demonstrate high predictive accuracy, the LIME explained that the fourth learner (Gradient Boosting) added redundant or contradictory feature importance in some cases. Say, such a feature as http-status-code may be used in both positive and negative forms in similar classifications. Here, the implication is that although diversity in ensembles is a good thing, there is a point at which the complexity becomes counterproductive due to a loss of interpretability, associated with a blurring of the explanatory signal.

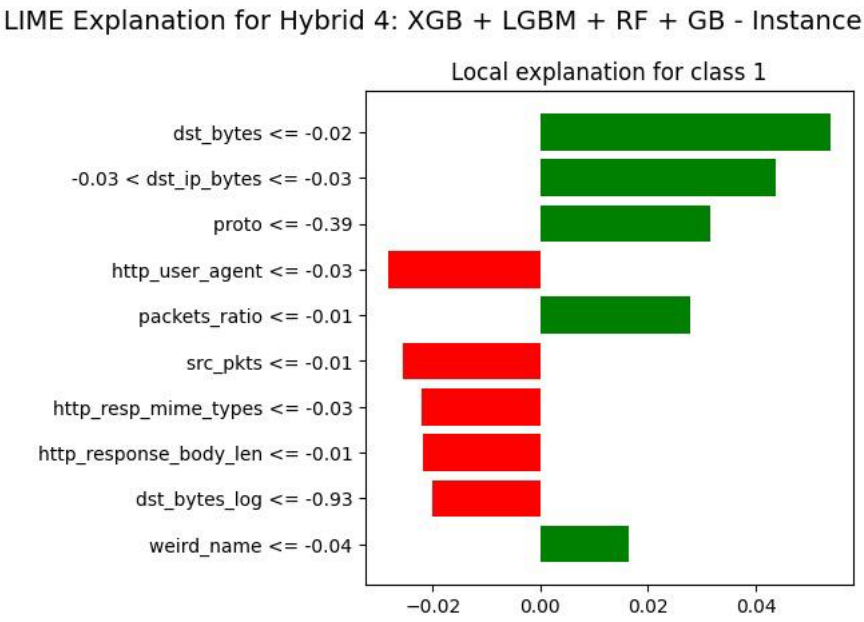

Figure 6.11: LIME Explanation for Hybrid 4: XGB + LGBM + RF + GB - Instance 1

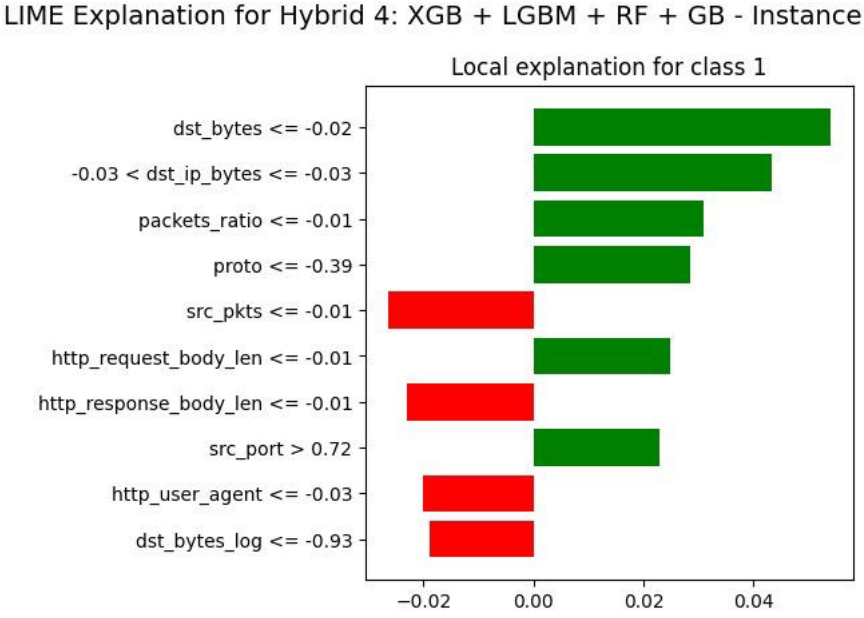

Figure 6.12: LIME Explanation for Hybrid 4: XGB + LGBM + RF + GB - Instance 2

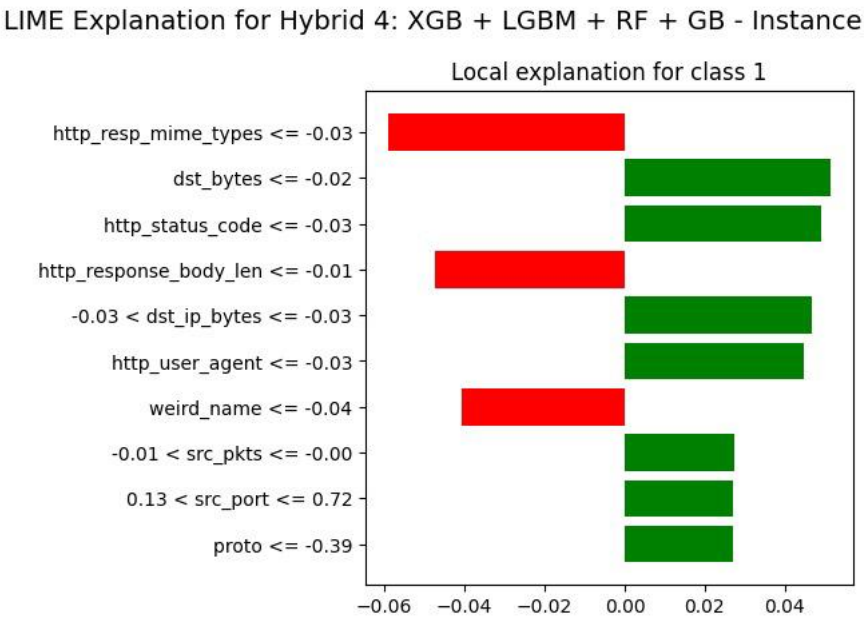

Figure 6.13: LIME Explanation for Hybrid 4: XGB + LGBM + RF + GB - Instance 3

6.7 Significance of Using LIME in Our Study

Though high detection accuracy is a requirement in any type of Intrusion Detection System (IDS), it is not the only requirement independently, particularly in a decision environment in which the decisions must be audited, human decision analysts act on the decisions, or real-time implementation. In these types of high-security areas, interpretability is just as crucial because it enables human beings who run the operations to comprehend and trust why an alert appeared. It is a fact that certain traditional machine learning models, e.g. Decision Trees or Logistic Regression, are interpretable by nature. Their choice mechanisms can be pictured or described with model coefficients or rules easily. Nonetheless, they do not work well with detecting multi-pattern intrusion behavior in current traffic as they have a relatively weak expressiveness and lower accuracy of detection.

To limit these drawbacks, in our study, we employed the more robust set of ensemble and hybrid models namely: XGBoost, LightGBM, Random forest, and their combination. Although there is a marked improvement in classification performance of these models, they are fundamentally not transparent, and most so when it is used in soft-voting ensembles where predictions are the aggregate output of multiple learners. Even though such models generate global measures of feature importance, these measures do not allow arguing why one instance has been identified as an injection or an XSS attack. In order to overcome this gap we ran LIME (Local Interpretable Model-Agnostic Explanations) onto our framework. LIME generates predictions at a local level by approximating the model in the vicinity of a particular prediction, showing which features most aided that decision, and whether the impact of that feature was positive or negative, as well as how sure the model was of that classification decision.

We have Lime applied to our four most vital models which included: the base XG-

Boost, and three hybrid ensembles that included Hybrid 2 (XGBoost + LightGBM), Hybrid 3 (XGBoost + LightGBM + Random Forest), and Hybrid 4 (XGBoost + LightGBM + Random Forest + Gradient Boosting). By this, we could compare and confirm the decision-making behavior of each model through more than just a numerical performance. As an example, LIME helped us understand how some features like http_response_body_len, packets_ratio, and dns_rcode were uniformly utilized by our best models in a fashion consistent with both domain knowledge and signature patterns. This understanding was necessary in trying to differentiate between actual learning and mere coincidence. As such, LIME was not merely a visualisation tool, but a fundamental validation method that has both ensured accuracy of our models, and provided interpretability, thus facilitating the deployment and accountability of decisions in cyber security practice.

6.8 Comparative Analysis

Comparative Analysis Table: Our Proposed IDS vs. The Existing on Research (Yes = Available, No = Not Available)

Research Study	**Model Type and Architecture**	**Explainability Technique**	**Attack Coverage**	**Data Handling**	**Real-time Capability**	**Drawbacks**
Sowmya et al. (2023)	SVM, KNN, classical ML & DL models	No	Standard Attacks only	No leakage, verify, out of balance	No	Absence of explainability, no hybrid modeling
Tcydenova and Kim (2021)	Deep Neural Network	Yes (SHAP)	Only Adversarial testing	No leakage management	No	Weakness of resistance to malicious traffic
Poudyal et al. (2020)	AIRaD Hybrid	No	Ransomware only	Not generalized data	No	Small scope, explainability lacks

Sriram et al. (2022)	Classical ML (RF, DT, SVM)	Yes (SHAP, LIME)	Only Basic anomalies	Not balanced	No	Limited model diversity
Him et al. (2024)	AI and ML Integrated System	No	Real-time threats	Unknown leakage and balancing	Yes	No instance level explainability
Our Work (2025)	Hybrid Ensemble (XGBoost + LightGBM + RF)	Yes (LIME)	XSS, Injection, DDoS, Scanning and Password.	Leakage addressed and Balanced	Ready to be deployed	None identified

Table 6.1: Comparison of Research Studies on Intrusion Detection Approaches

Why Our work is unique:

- Integrates explainability (LIME) with hybrid ensemble architecture, something that is uncommon to the available literature.

- Covers the various categories of the present-day attacks such as XSS, Injection, and adversarial patterns.

- Performs leakage mitigation and class balancing in the preprocessing to make it more valid.

- Exblue suggests both high detection performance (F1 = 97.25%) and transparency, which helps its use.

- No severe shortcoming such as other works- comprehensive, explanatory, and non-attackable coverage.

Chapter 7

Conclusion

A complex evolution of cyber threats has rendered the AI-controlled Intrusion Detection Systems (IDS) a crucial feature of the current network security. Although machine learning and deep learning-based models can improve the accuracy of threat detection, they are easily affected by competitive attacks and are not very interpretable. The current study aims at solving these limitations as it incorporates Explainable AI (XAI), specifically LIME, into the IDS model to enhance trust and model transparency. We have tried different kinds of traditional and ensemble models, such as XGBoost, LightGBM, and hybrids, to compare them to each other in standard evaluations. Our results indicate that a combination of interpretable tools and state of the art models is not only more accurate in detection but it also increases the reliability of the system. This would be a viable and scalable methodology of constructing secure transparent IDS. In future, the work can research to resilient models in real-time flexibility and other XAI techniques that could enhance model resilience further.

Bibliography

[1] G. Kumar, "The use of artificial intelligence-based techniques for intrusion detection: A review," *Open Athens*, Sep. 2010. [Online]. Available: https://link.springer.com/article/10.1007/s10462-010-9179-5.

[2] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, CA, USA: ACM, 2016, pp. 785–794. DOI: 10.1145/2939672.2939785.

[3] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should i trust you?": Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, CA, USA: ACM, 2016, pp. 1135–1144. DOI: 10.1145/2939672.2939778.

[4] G. Ke, Q. Meng, T. Finley, *et al.*, "Lightgbm: A highly efficient gradient boosting decision tree," in *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS)*, Curran Associates, Inc., 2017, pp. 3146–3154. [Online]. Available: https://papers.nips.cc/paper_files/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf.

[5] H. H. Patel and P. Prajapati, "A comprehensive study on decision tree algorithms and their evaluation," *International Journal of Computer Sciences and Engineering*, vol. 6, no. 10, pp. 1–6, Oct. 2018. [Online]. Available: https://www.ijcseonline.org/pub_paper/14-IJCSE-04947.pdf.

[6] B. M. Khammas, "Ransomware detection using random forest technique," *ScienceDirect*, vol. 6, no. 4, pp. 325–331, Dec. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405959520304756.

[7] S. Khan, E. Sivaraman, and P. B. Honnavall, "Performance evaluation of advanced machine learning algorithms for network intrusion detection systems," *ResearchGate*, pp. 51–59, Apr. 2020. [Online]. Available: https://www.researchgate.net/publication/340522789_Performance_Evaluation_of_Advanced_Machine_Learning_Algorithms_for_Network_Intrusion_Detection_System.

[8] N. Moustafa, T. M. Booij, I. Chiscop, E. Meeuwissen, Z. Tari, and F. T. H. den Hartog, *TON-IoT telemetry dataset: Heterogeneous telemetry from iot/iiot, os, and network sources for intrusion detection*, https://research.unsw.edu.au/projects/toniot-datasets, [Dataset], UNSW Canberra, 2020.

[9] S. Poudyal and D. Dasgupta, "Ai-powered ransomware detection framework," *ResearchGate*, Dec. 2020. [Online]. Available: https://www.researchgate.net/publication/346577369_AI-Powered_Ransomware_Detection_Framework.

[10] S. Zeadally, E. Adi, Z. Baig, and I. A. Khan, "Harnessing artificial intelligence capabilities to improve cybersecurity," *IEEE*, 2020, [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8963730/authors#authors.

[11] M. Aljabri, S. Aljameel, R. M. A. Mohammad, *et al.*, "Intelligent techniques for detecting network attacks: Review and research directions," *ResearchGate*, Oct. 2021. [Online]. Available: https://www.researchgate.net/publication/355591752_Intelligent_Techniques_for_Detecting_Network_Attacks_Review_and_Research_Directions.

[12] S. Tatineni, "Machine learning approaches for anomaly detection in cybersecurity: A comparative analysis," *International Journal of Computer Engineering and Technology*, vol. 12, no. 2, pp. 42–50, 2021. [Online]. Available: https://www.researchgate.net/publication/377434418_MACHINE_LEARNING_APPROACHES_FOR_ANOMALY_DETECTION_IN_CYBERSECURITY_A_COMPARATIVE_ANALYSIS.

[13] E. Tcydenova and T. W. Kim, "Detection of adversarial attacks in ai-based intrusion detection systems using explainable ai," *ResearchGate*, Sep. 2021. [Online]. Available: http://hcisj.com/data/file/article/2021091501/11-35.pdf?ckattempt=2.

[14] B. Dash, M. F. Ansari, P. Sharma, and A. Ali, "Threats and opportunities with ai-based cyber security intrusion detection: A review," *International Journal of Software Engineering & Applications (IJSEA)*, vol. 13, no. 5, Sep. 2022. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=43232581.

[15] S. Huh, S. Cho, J. Choi, S. Shin, and H. Lee, "A comprehensive analysis of today's malware and its distribution network: Common adversary strategies and implications," *IEEE Access*, vol. 10, Apr. 2022. [Online]. Available: https://www.researchgate.net/publication/360266102_A_Comprehensive_Analysis_of_T.

[16] N. Y. Jakka and M. F. Ansari, "Artificial intelligence in terms of spotting malware and delivering cyber risk management," *Journal of Positive School Psychology*, Apr. 2022. [Online]. Available: https://journalppw.com/index.php/jpsp/article/view/3522.

[17] H. Li, "Explainable intelligence-driven defense mechanism against advanced persistent threats: A joint edge game and ai approach," *IEEE Xplore*, Apr. 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9627773.

[18] A. Rawal, J. McCoy, D. B. Rawat, B. M. Sadler, and R. S. Amant, "Recent advances in trustworthy explainable artificial intelligence: Status, challenges, and perspectives," *IEEE Transactions on Artificial Intelligence*, vol. 3, no. 6, pp. 852–866, Dec. 2022. DOI: 10.1109/TAI.2021.3133846. [Online]. Available: https://ieeexplore.ieee.org/document/9645355.

[19] G. K. Sriram, "Explainable ai for intrusion detection systems," *IEEE Xplore*, 2022. [Online]. Available: https://ieeexplore.ieee.org/document/10073356.

[20] A. Fathia, "Defending against adversarial attacks in ai-powered cybersecurity: A comprehensive exploration," *ResearchGate*, Jul. 2023. [Online]. Available: https://www.researchgate.net/profile/Ademola-Fathia/publication/380343243_Defending_Against_Adversarial_Attacks_in_AI-Powered_Cybersecurity_A_Comprehensive_Exploration/links/663642517091b94e93ef23bd/Defending-Against-Adversarial-Attacks-in-AI-Powered-Cyber.

[21] J. Owen, "Fortifying cybersecurity: Exploring defensive strategies against adversarial attacks on ai-driven systems," *ResearchGate*, 2023. [Online]. Available: https://www.researchgate.net/publication/380295975_Fortifying_Cybersecurity_Exploring_Defensive_Strategies_Against_Adversarial_Attacks_on_AI-Driven_Systems.

[22] M. Schmitt, "Securing the digital world: Protecting smart infrastructures and digital industries with ai-enabled malware and intrusion detection," *Journal of Industrial Information Integration*, vol. 36, p. 100 520, Dec. 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2452414X23000936.

[23] T. Sowmya and E. M. Anita, "A comprehensive review of ai based intrusion detection system," *Measurement. Sensors*, vol. 28, p. 100827, 2023. DOI: 10.1016/j.measen.2023.100827.

[24] M. Vishwakarma and N. Kesswani, "A new two-phase intrusion detection system with naïve bayes machine learning for data classification and elliptic envelop method for anomaly detection," *ScienceDirect*, vol. 7, Jun. 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2772662223000735.

[25] I. Him, "Leveraging artificial intelligence and machine learning for real-time threat intelligence: Enhancing incident response capabilities," *ResearchGate*, 2024. [Online]. Available: https://www.researchgate.net/publication/380295960_Leveraging_Artificial_Intelligence_and_Machine_Learning_for_Real-Time_Threat_Intelligence_Enhancing_Incident_Response_Capabilities.

[26] S. Muneer, U. Farooq, A. Athar, M. R. R. Raza, T. M. Ghazal, and S. Sakib, "A critical review of artificial intelligence-based approaches in intrusion detection: A comprehensive analysis," *Journal of Engineering*, Apr. 2024. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1155/2024/3909173.

[27] M. Manninen, "Using artificial intelligence in intrusion detection systems," *ResearchGate*, [Online]. Available: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=07f0dedff3761fd6aa6b52b3a8c35a280b4b7b1a.