University Of Thessaly

School of Science

Department Of Computer Science and Biomedical Informatics

Adversarial Machine Learning attacks against ML based intrusion detection systems

Nikolaos Nikas

Thesis

Advisor

Georgios Spathoulas

Laboratory Teaching Staff

Lamia, 17/02/2025

Με ατομική μου ευθύνη και γνωρίζοντας τις κυρώσεις [(1)], που προβλέπονται από της διατάξεις της παρ. 6 του άρθρου 22 του Ν. 1599/1986, δηλώνω ότι:

1. *Δεν παραθέτω κομμάτια βιβλίων ή άρθρων ή εργασιών άλλων αυτολεξεί* ***χωρίς να τα περικλείω σε εισαγωγικά*** *και χωρίς να αναφέρω το συγγραφέα, τη χρονολογία, τη σελίδα. Η αυτολεξεί παράθεση χωρίς εισαγωγικά χωρίς αναφορά στην πηγή, είναι λογοκλοπή. Πέραν της αυτολεξεί παράθεσης, λογοκλοπή θεωρείται και η παράφραση εδαφίων από έργα άλλων, συμπεριλαμβανομένων και έργων συμφοιτητών μου, καθώς και η παράθεση στοιχείων που άλλοι συνέλεξαν ή επεξεργάσθηκαν, χωρίς αναφορά στην πηγή. Αναφέρω πάντοτε με πληρότητα την πηγή κάτω από τον πίνακα ή σχέδιο, όπως στα παραθέματα.*

2. *Δέχομαι ότι η αυτολεξεί* ***παράθεση χωρίς εισαγωγικά****, ακόμα κι αν συνοδεύεται από αναφορά στην πηγή σε κάποιο άλλο σημείο του κειμένου ή στο τέλος του, είναι αντιγραφή. Η αναφορά στην πηγή στο τέλος π.χ. μιας παραγράφου ή μιας σελίδας, δεν δικαιολογεί συρραφή εδαφίων έργου άλλου συγγραφέα, έστω και παραφρασμένων, και παρουσίασή τους ως δική μου εργασία.*

3. *Δέχομαι ότι υπάρχει επίσης περιορισμός στο μέγεθος και στη συχνότητα των παραθεμάτων που μπορώ να εντάξω στην εργασία μου εντός εισαγωγικών. Κάθε μεγάλο παράθεμα (π.χ. σε πίνακα ή πλαίσιο, κλπ), προϋποθέτει ειδικές ρυθμίσεις, και όταν δημοσιεύεται προϋποθέτει την άδεια του συγγραφέα ή του εκδότη. Το ίδιο και οι πίνακες και τα σχέδια*

4. Δέχομαι όλες τις συνέπειες σε περίπτωση λογοκλοπής ή αντιγραφής.

Ημερομηνία: 17/2/2025

Ο – Η Δηλ.

(Υπογραφή)

(1) «Όποιος εν γνώσει του δηλώνει ψευδή γεγονότα ή αρνείται ή αποκρύπτει τα αληθινά με έγγραφη υπεύθυνη δήλωση του άρθρου 8 παρ. 4 Ν. 1599/1986 τιμωρείται με φυλάκιση τουλάχιστον τριών μηνών. Εάν ο υπαίτιος αυτών των πράξεων σκόπευε να προσπορίσει στον εαυτόν του ή σε άλλον περιουσιακό όφελος βλάπτοντας τρίτον ή σκόπευε να βλάψει άλλον, τιμωρείται με κάθειρξη μέχρι 10 ετών.

Thesis

Adversarial Machine Learning attacks against ML based intrusion detection systems

Nikolaos Nikas

Committee:

Georgios Spathoulas, Laboratory Teaching Staff (advisor)

Theodoros Tzouramanis, Assistant Proffesor

Athanasios Kakarountas, Associate Proffesor

Table of Contents

Table of Figures

Abstract

Intrusion Detection Systems (IDS) are crucial in safeguarding computer networks from cyber threats. As cyber-attacks become increasingly sophisticated, traditional IDS approaches struggle to keep up with evolving threats. This study presents an approach using the Random Forest algorithm for intrusion detection, leveraging its ensemble learning capabilities to enhance classification accuracy. The methodology includes dataset preprocessing, feature selection, model training, and evaluation, with a focus on optimizing performance through hyperparameter tuning.

The Random Forest classifier, known for its robustness and ability to handle large datasets, is employed to distinguish between normal and malicious network traffic. By aggregating multiple decision trees, the algorithm enhances prediction stability and minimizes overfitting. The study utilizes publicly available intrusion detection datasets, ensuring a diverse range of attack scenarios are considered. Various performance metrics, including accuracy, precision, recall, and F1-score, are employed to evaluate the model's effectiveness.

A key aspect of this research is the exploration of adversarial bypassing attacks, a critical vulnerability in IDS. Attackers often attempt to evade detection by introducing subtle perturbations to input data, leading to misclassification. This study implements an adversarial evasion mechanism that selectively modifies specific features to bypass detection while maintaining classification reliability. The approach leverages gradient-based perturbations with momentum and controlled noise injection to simulate real-world adversarial scenarios. Experimental results demonstrate that while adversarial attacks can significantly degrade IDS performance, targeted defenses can mitigate their impact and maintain high detection accuracy.

By integrating adversarial attack resilience into machine learning-based IDS, this research highlights the need for continuous adaptation in threat detection methodologies. The findings emphasize the importance of developing more robust and adaptive cybersecurity solutions to counter emerging evasion strategies and ensure network security in dynamic threat environments.

1. Introduction

An **Intrusion Detection System** is an indispensable security technology element of the current digital security system that allows for defining and preventing unauthorized activities and potential threats and attacks against networks, devices, and information. These are constantly in operation as analyzers, detectors, and counter-action-takers for any prohibited activities within an organization's IT structure. This is quickly amplified by the current world, where there is accelerated advancement and interconnectivity of digital commodities and services that have made the threat and attack forms larger and more comprehensive, thus requiring deeper barriers. Hence, the organization relies on IDS to maintain data integrity, confidentiality, and accessibility to the organizations and other stakeholders in organizations (Abdulganiyu, 2024).

An IDS is a core security technology whereby optimum protection is offered against unauthorized attempts as well as other potential threats within a network device or information system. IDS solutions are always monitoring traffic flows, identifying abnormal activity and react to cease intrusions. From the emergence and growth of digital technologies and connectivity of devices, the avenues through which an IDS must protect data from being edged, stolen, or made unavailable are; continually growing.

IDS can be categorized into two major categories namely the host based IDS (HIDS) and the network based IDS (NIDS). As for HIDS, it is aimed at the device level to distinguish counting anomalies in the file access and changes in the system log, among others. Rather, NIDS remains focused on the network traffic analysis to detect any form of anomaly that might be a sign of an intrusion. The conventional IDS techniques mostly used are the signature based detection that compares traffic flows against attack signatures and anomaly based detection, which looks for the

deviation from established norms. Nevertheless, such approaches do not work well for identifying new and emerging threats in the cyberspace.

The Rise of Machine Learning in IDS

Machine learning (ML) and artificial intelligence (AI) have significantly enhanced IDS capabilities by enabling automated pattern recognition and adaptive threat detection. Unlike rule-based systems that require frequent manual updates, ML-based IDS learn from historical data, improving their ability to detect zero-day attacks and novel intrusion techniques. Algorithms such as Random Forest, Support Vector Machines (SVM), and Neural Networks have demonstrated strong performance in identifying malicious activities with higher accuracy and lower false positive rates.

The introduction of ML in IDS has also facilitated large-scale data processing, making it possible to analyze vast amounts of network traffic in real time. Feature selection techniques, such as Recursive Feature Elimination (RFE) and feature importance ranking, help identify the most relevant indicators of intrusion, improving the efficiency and interpretability of IDS models. Despite these advancements, ML-based IDS remain vulnerable to adversarial attacks, which are carefully crafted perturbations designed to evade detection.

Adversarial Bypassing Attacks in IDS

One of the most pressing challenges in modern IDS is the threat of adversarial bypassing attacks. Attackers manipulate network traffic features by introducing small, imperceptible changes that mislead the IDS into classifying malicious traffic as benign. These adversarial perturbations exploit the weaknesses in ML models, causing them to misclassify or fail to detect intrusions altogether.

Several techniques are commonly used in adversarial evasion attacks, including:

- Gradient-based perturbations: Attackers compute the gradient of the IDS model's decision function and introduce perturbations along the most sensitive features.
- Feature-space transformations: Modifying packet timing, encryption patterns, or payload structures to resemble normal traffic while retaining malicious intent.
- Random noise injection: Introducing small variations to bypass IDS without significantly altering the underlying attack strategy.

To address these vulnerabilities, this study incorporates an adversarial attack bypassing mechanism within the IDS framework. By selectively modifying specific network features (e.g., packet size, protocol type, and flow duration), the model is tested against adversarial conditions to assess its resilience. The approach employs momentum-based gradient perturbations and controlled noise injection to evaluate IDS robustness. Experimental results indicate that traditional ML-based IDS suffer performance degradation under adversarial conditions, while targeted adaptation strategies can mitigate evasion attempts and maintain high detection accuracy.

IDS can be divided broadly into two categories which are **host-based intrusion detection systems** and network based intrusion detection systems. **HIDS is concerned with monitoring** activity within the confines of a single device, most commonly on a server, workstation, or endpoint. In addition, it maintains a check on log files and other patterns which constitute operations of a possibly attacked system. NIDS works at the level of an entire network and monitors traffic across the entire networks for any unusual activity that could signal an intrusion. These systems employ different techniques of detection, such as signature-based detection, where the system looks for patterns of identified attacks to seek violations of security, and anomaly-based detection, which

looks out for the probability or possibility of an attack that has not even been recognized before (Abed, 2024).

The effectiveness of IDS depends on its capability to detect new threats while producing the least possible false positives and false negatives, which are wrong detections of bad traffic and failed detections of bad traffic, respectively. Many traditional IDSs, which were developed as rule-based detection systems, are usually slow in responding to the continually changing methods used by the attackers. Such systems, while being equipped to identify and stop known threats, are not amenable to detecting new forms of attack or new techniques of an already known one (Abid, A., Jemili, 2024).

Due to the previous limitations of these systems, newer IDS today integrates structures and technologies like ML and AI. These technologies assist in the improvement of the capability of IDS by allowing it to analyze previous data and learn about the patterns of threats by providing it with advanced features. The incorporation of ML and AI has brought radical changes in the cybersecurity domain by introducing a more adaptive, timely, and automated approach to solving problems in contrast to the previously traditional approach to threat protection as the threat factors have grown bigger and more complicated with a considerable volume (Ables et al., 2024).

Another general contribution of ML to the advancement of IDS is that the technology can identify patterns and anomalous data within large and complex data sets. The traditional systems work well with a rule-based system with predetermined signatures and patterns; the problem with this approach is that it is not suitable for the new attacks that have not been accounted for in the system's design. However, ML-based IDS can learn from past network traffic, where, like other IDS, it will not be able to identify new kinds of attacks and the strategies adopted by the attackers.

The effectiveness of machine learning also makes the system capable of adapting to the latest developing data to have better and more advanced detection capacity (Ahmed, Q.O., 2024).

The machine learning component performs well in dealing with big data, which is quite a significant issue in the cybersecurity environment. Since the cybersecurity systems produce real-time logs, network traffic and event information, it is almost impossible to process and detect threats oneself or with regular systems. The kind of data that turns up in compliance often can be handled by ML algorithms, including Random forest and SVM and neural networks, to make predictions because of its size and the fact that sometimes subtleties that are obvious to even simple rule-based systems might not be immediately apparent to human analysts (Akhiat, 2024).

However, one of the most critical concerns when applying IDS with machine learning is the problems that are associated with false positives and false negatives; that is a major drawback of traditional IDS that makes them less effective in their intended operations. There is no doubt that via the application of supervised learning techniques, the training of the ML models can quickly be done on different labeled datasets, which comprise both normal and malicious traffic patterns, thus enhancing the identification skills of the correct traffic from the wrong traffic. Concretely, clustering, which is unsupervised learning, can be used to identify unknown or new types of attacks that have not been labeled earlier (Alalhareth, 2024).

Another IDS capability that the integration of ML has brought about is that it provides a specific probability that a particular type of threat will happen. These schemes operate on the records of their past occurrences, show recent occurrences, and prevent or preempt further advancement of malice before they execute a mass attack. This feature significantly bolsters IDS's capacity for anticipation and provides organizations with a factor through which risks can be bolstered prior to an attack (Al-Ambusaidi, 2024).

In today's world with rapidly evolving threats and increasing threat actors, cyber security professionals are often faced with countering a number of aggressive attempts to threaten the integrity, confidentiality, and availability of information systems. These threats appear in different forms, including viruses, worms, Trojans, phishing scams that aim to gain one's login details or even financial details, and DoS or DDoS attacks, which expose networks with vast amounts of traffic to make them inaccessible. Another type of attack is APT, where attackers infiltrate a target and remain there for an extended period to steal data from the system; the other one is zero-day attack, which targets vulnerabilities that are unknown to vendors, thus making its prevention difficult (Aldeen et al., 2025).

Besides those kinds of **cyberattacks**, there are general weaknesses that are inherent inside an organization and can be invoked by attackers. If a program fails to use strong passwords or updates passwords often, it bypasses a security point. This type of software is dangerous as well because, with vulnerabilities, systems remain defenses open for hackers to attack. Secondly, any misconfigured systems, which can be networks, databases, applications, and others, mean attackers have an entry point to a network, and therefore, the system is vulnerable. Also, the employee or contractor's **malicious threats** are always possible since these insiders may intentionally or non-intentionally damage the system (Ali, B.S., 2024).

These **cyber-threats** could be very disastrous to organizations if they are attacked. Some of the consequences of such threats include the following: Financial losses: These are some of the most significant risks, which may lead to considerable losses and other adverse effects. Botnets, for instance, may entail substantial downtime and result in organizations having to pay ransoms in order to access their systems. At the same time, in **data breaches**, customer information is likely

to be leaked, customers are likely to lose trust, organizations launch into legalities, and in the long run, they suffer reputational loses.

This paper aims to **design and implement** an IDS based on the Random Forest classifier that will be able to detect and distinguish real threats within a network. This approach seeks to address the shortcomings associated with traditional IDS, like lack of bankruptcy to the system, producing false alarms, and inability to detect new attacks. The incorporation of machine learning will lead to an increase in the detection of malicious behavior, increased speed, and reliability in the defense mechanism of the system (Yousafzai, A., 2024).

The main objectives of this project include:

1. **Data Processing and Preparation:** Cleaning the data given for this project so that they are in a format suitable for training and validation.
2. **Model Development:** Designing a Random Forest-based classification model with great potential for identifying regular and suspicious activity on a network.
3. **Validation:** Evaluating the model's effectiveness based on basic metrics such as accuracy, precision, recall, and F1 score.
4. **Interpretation and Visualization:** Presenting information in the form of interpretations and visuals will be very helpful when making new decisions and comprehending the system's operations.
5. **Scalability:** The system's capacity for expansion and adjustment to various network topologies increases the possibility of applying this approach to different large network systems (Aljehane, 2025).

As such, these objectives help to enrich the cybersecurity frameworks as well the development of the machine learning for more effective, versatile, and highly functional Intrusion Detection Systems which may cater with new threats encountered in the digital world.

2. Literature Review

Intrusion Detection Systems (IDS) have been extensively studied in cybersecurity research as a fundamental layer of defense against unauthorized access, malware, and network intrusions. This section provides an in-depth review of traditional IDS approaches, the emergence of machine learning-based IDS, and the growing challenge of adversarial attacks that attempt to bypass these security measures.

2.1 Traditional Approaches to Intrusion Detection

Historically, IDS relied on **signature-based detection** and **rule-based detection** mechanisms to identify cyber threats. These methods have been effective in detecting known attack patterns but suffer from several limitations, particularly in identifying new and evolving threats.

2.1.1 Signature-Based Detection

Signature-based IDS compares incoming traffic with a database of predefined attack signatures. This method is widely used in commercial IDS solutions, such as **Snort** and **Suricata**, which maintain extensive rule sets for detecting known exploits. The main advantage of this approach is its **high accuracy for well-documented attacks** and its **low false positive rate**.

However, signature-based detection suffers from the following drawbacks:

- **Inability to detect novel attacks**: It can only recognize attacks that match known signatures. Zero-day attacks and polymorphic malware evade detection by modifying their payloads.
- **Frequent signature updates**: The system requires constant manual updates to include newly discovered attack patterns.

- **High maintenance costs**: Security analysts must continuously refine and expand signature databases, which is resource-intensive.

2.1.2 Rule-Based Detection

Rule-based IDS (also called expert systems) defines intrusion patterns using **predefined heuristics**. Examples include defining a rule such as:

"If a single IP sends more than 1000 requests in one minute, flag as potential DoS attack."

Rule-based systems, such as those implemented in **firewall security policies**, provide fast detection but are prone to **false positives** when normal network activity matches predefined rules. Moreover, attackers often develop **evasion techniques** to bypass static rules, leading to **high false negatives** (missed intrusions).

2.1.3 Anomaly-Based Detection

Anomaly-based IDS aims to detect deviations from normal behavior rather than relying on predefined signatures. These systems employ statistical models to define a baseline of "normal" network traffic and flag unusual deviations as potential threats.

Advantages of Anomaly-Based IDS:

Can detect **zero-day attacks** since it does not rely on predefined signatures. More **adaptive** to changing attack patterns.

Challenges of Anomaly-Based IDS:

High false positive rate: Unusual but legitimate traffic patterns can trigger alarms.
Difficulty in defining “normal” behavior: Large-scale networks experience fluctuations, making it hard to establish a strict baseline.

Susceptibility to adversarial attacks: Attackers can manipulate the system's learning phase to disguise malicious activity as normal.

Due to these limitations, cybersecurity researchers have turned to **machine learning (ML)-based IDS**, which improves detection accuracy by automatically learning attack patterns from data.

2.2 Traditional Approaches

Behind the acronym Intrusion Detection System (IDS), one can trace the development of IDS as a significant component of cybersecurity throughout the years to meet the continuously growing and developing threats to global information systems. At the early stage, IDS mainly depended on signature detection and a rule-based system. Based on the domain, signature-based detection engaged in the identification of attack patterns called signatures that were obtained from known attacks, in the same way antic viruses work with viruses. Though this method was rather successful against previous threats, it had one seemingly obvious drawback: new and unknown types of attacks could not be recognized, or rather, the latest form of intrusion was not uncovered until a certain signature for it was entered into the system (Almehdhar, 2024).

Rule-based detection, another early approach, is accurately named since it entails the formulation of actual preeminent rules or heuristics using models that were learned from past attacks. These rules were then applied to the network traffic, and any observation of rules previously set out in the guidelines that did not conform to the defined norms was considered suspicious. Similar to the methods based on the use of a signature, rule-based methods had a disadvantage common to them: their effectiveness critically depended on certain templates and required constant updates due to the changing nature of the threats. Since new attack vectors and attack tactics were continuously being identified, security administrators had to keep updating such rules, which was a time-consuming and impractical process as new threats emerged in the evolving cyber threat landscape.

Thus, as a result of the shortcomings of such systems as signature-based and rule-based, the concept of anomaly-based detection appeared. Anomaly-based IDS deals with comparing some recognized standards or patterns of activity of a system or network in an attempt to detect the presence of an intrusion. This proved to be a far better approach to the generally traditional signature-based model in a way that could isolate PSs that are unknown but render different activities that the signature model would not be able to differentiate. Nevertheless, it had its problems. The first problem was the high false-positive rate, as many perfectly safe activities like new applications or rather rare but legitimate user actions were considered suspicious. One of the significant difficulties was correctly describing which activity could be regarded as usual for a complex, challenging to define due to evolving dynamics resulting from such factors as the deployment of new equipment or application. This challenge results in many misclassifications and floods the security team with many alerts, making the IDS very less effective (Alqahtany, 2024).

A major disadvantage of all three broad classes of IDS, such as the signature-based IDS, the rule-based IDS, and the anomaly-based IDS, is that the IDS were not dynamic systems. Developing a list of new signatures and frequently occurring rule changes as well as threshold settings was also a continuous one as the older and new threats needed to be detected by the system. Network conditions became more extensive with increased scales, and IDS failed to cope with large volumes of information collected by modern networks. This resulted in performance bottlenecks and lengthened time necessary to alert of possible intrusions; therefore, these systems are not very efficient to use in real-time settings. The rise in the sophistication of the attacks and the exponentially increasing amount of network traffic that has to be processed necessitated improvements in IDS technologies (Amaouche, S., 2024).

In response to these limitations, hybrid IDS systems are included in their design. These systems integrate the features from both the signature-based and anomaly-based systems as a way of providing a better solution in an IDS. Through the integration of predefined patterns with learning capability over the network behaviors, the hybrid system was intended to provide better or improved detection capability and coverage. Though the hybrid IDS was a progressive advancement over traditional techniques, it proved to have some drawbacks in terms of flexibility and efficiency in case a further and continuous threat was in the network. With time, new threats emerged and more of the previous approaches remained incapable of detecting APTs, complex multi-stage attacks, and other converging attacks that employed the new complex evasion techniques (Andreica, T., 2024).

However, such an IDS needs to be based on ML and AI since these are capable of dynamically adapting to newer threats, having been trained on past data. Unlike other systems, there is no use of signatures or rules in the machine learning-based IDS; instead, there is the use of large volumes of data to build a set of models that will be able to recognize sophisticated patterns of attack associated with cyber threats. This has the capability of making IDS adapt over time with the primary purpose of learning from previous attacks and enhancing the model with newer data. The integration of machine learning-based IDS also resolves most of the problems related to legacy systems, such as non-adaptive nature, high rates of false positives, and the inability to identify new threats by the system (Arreche, O., 2024).

Indeed, IDS that use machine learning algorithms have easily matched the ability of human security analysts to analyze large data flows and outperformed them inshore and in real-time analysis. The present networks produce immeasurable amounts of traffic data ranging from network traffic logs right through to system and application logs which overwhelm traditional IDS

in terms of the processing and analysis of data. Some algorithms like Random forest, Support Vector Machines (SVM), and Neural networks work very well in processing large datasets for patterns, which might be difficult for even humans or any rule-based system to find out. This being the case, these models are able to read from the data and find out those conducts that are anomalous and the possibility of zero-day attacks, as well as have the ability to actively learn and update themselves with the changing threats without the need for constant human interference (Arsalan, 2024).

Two other advantages that are associated with the use of machine learning-based IDS are: Another considerable advantage of IDS that employ machine learning techniques is that it is less prone to false positives as well as false negatives. This way, the machine learning algorithms can learn on the labeled datasets and identify whether the network activities are normal or malicious with higher credibility. Supervised techniques, also referred to as labeled techniques, involve the use of data with labels, thus making the models improve their decision-making due to attack experiences. Moreover, clustering and anomaly detection methods are able to discover new patterns of attack profiles and new types of attacks for which no labels have been given. These features make machine learning-based IDS much more efficient in the identification of a wide range of threat types as well as in relieving security personnel (Asaju, B.J., 2024).

Furthermore, the use of machine learning in IDS systems has also resulted in what can be termed as predictive IDS, which is one that is capable of predicting an attack before it takes place. It is one of the best ways through which an organization can prevent or minimize risks before they escalate to the level of a cyber-attack. Through the analysis of past data and its consecutive changes, team learning models can predict and make the clients aware of possible loopholes in their systems so that they can be dealt with before being exploited by attackers.

The incorporation of machine learning and Artificial Intelligence in IDS systems has made them closer to being part of modern cybersecurity frameworks that can evolve on their own. They are not any more of the manual type that needs constant modifications on rules or signatures; instead, these new systems can also learn on their own, making them more complete and efficient in terms of defending a computer against various types of attacks. Given the current trends in the world of cyberspace, this paper intends to prove that the use of machine learning in IDS is going to be vital to help organizations adapt to the changing face of cyber threats. This shift to data-driven adaptive systems is essential in an environment where threats are sophisticated and often obscure, thus enabling organizations to implement one of the most effective solutions to secure their networks and systems against the daily growing threats. Given the fact that the IDS will become more sophisticated and extensive in the future, machine learning and AI are the primary tokens that will lead to the next generation of IDS that is proactive, adaptive, and efficient (Bouke, M.A., 2025).

2.2.3 Machine Learning-Based Methods

Today's automated IDS employs ML to improve the ability to identify new and unknown threats that rule-based systems cannot address on their own. In the last two decades, ML as technology has become somewhat effective in detecting malicious activities, and its early application included Classification techniques like Decision Trees and Naïve Bayes. These techniques were said to have used previous attack data to identify potent threats. However, as the network traffic load grew, the necessity of developing more accurate and efficient models appeared (Fenjan, A., 2024).

To overcome these issues of scalability, different methods of ensemble learning, such as Random Forest and Gradient Boosting, were developed. These improved the results of classification and minimized the number of false alarms because the results of many classifiers were combined. The

development of ML technology also moved from supervised learning that requires large sets of labeled data to other categories, such as unsupervised and semi-supervised learning. This is due to the fact that unsupervised methods such as clustering and anomaly detection were efficient in finding new threats that are unknown to the system since they do not require labels. A semi-supervised learning approach is aligned with the labeled and unlabeled datasets by enhancing detection efficiency without much tagging.

Deep learning has made this possible, making different models capable of handling sequential data to expose more intricate intrusion patterns. Algorithms such as Recurrent Neural Networks (RNNs) and Conventional Neural Networks (CNNs) have been seen to perform well when it comes to APT and zero-day attack detection. The adaptability of ML-driven IDS and constant learning qualify them for effective proactivity in threat detection; they are, therefore, prominently placed for use in the current threat detection mechanisms (Hazman, C., 2024).

2.4 Brief Overview of Random Forest Classifier

Random Forest is a practical and successful learning algorithm that is suitable for solving various classification problems. It falls in the category of Assembling or Bagging, where accuracy is improved by generating and averaging multiple models. Leo Breiman develops Random Forest; in this algorithm, during the training phase, a large number of decision trees are generated, and the result will be the final classification obtained by taking a majority vote from these trees.

The two techniques used in the algorithm include bootstrap aggregating and random feature choice. Bagging is carried out by using multiple decision trees, where each tree is trained using a boot strap sample of data with replacement to minimize overfitting. Furthermore, averaging the results also works on random subspaces of the features, so that most of the trees of the separated

nodes do not have many similar attributes. The final decision is made from the overall prediction by all the decision trees commonly through a voting system (Hizal, S., Akgun, D., 2024).

Random Forest is a widely used ensemble learning method applicable to classification and regression tasks. Its theoretical foundation is built upon principles of decision trees, bagging (bootstrap aggregating), and random feature selection, making it a powerful and versatile algorithm.

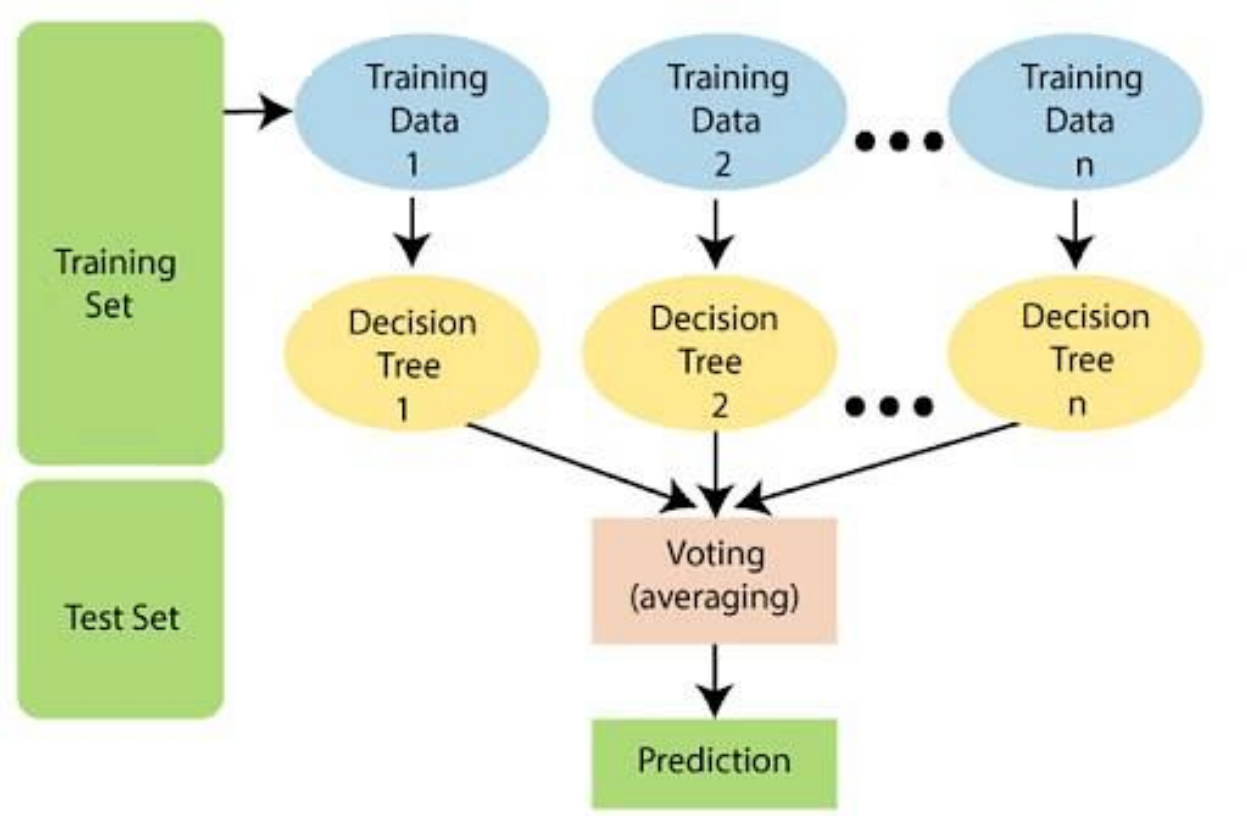

Figure 1Machine Learning Overflow

Figure 2Machine Learning Overflow

In other words, the approach forming the basis of Random Forest is an ensemble of decision trees. A decision tree is a technique of data partitioning that divides the data based on feature values. A tree map itself is a tree structure in which every node is a decision rule, and every node represents

an output class. This is true even when decision trees are combined because they are easily overfitting, mainly when used individually.

Bagging is the procedure that helps to increase the stability and accuracy of the developed machine-learning algorithms. As in the Random project, various bootstrap samples and random samples of the training data, drawn with replacement, are produced. Decision trees are trained on the training sample, and the result is the accounting of all trained decision trees, where a simple majority vote is usually present in the case of classification. This is because it reduces variance by making various decision trees relatively immune to noises within the set data, hence improving the model (Kalutharage, 2025).

Several key theoretical properties underpin the effectiveness of Random Forest. The law of large numbers reveals that as the number of trees grows, the predictions offered by the Random Forest approach the actual value given that data quantity and variety are enough. The bias-variance tradeoff is also important, as Random Forest significantly decreases the value of variance and has a low level of bias, which further increases its accuracy level. Further, out-of-bag (OOB) error estimation, which uses sample data not used in any tree in the construction of the model, provides an analysis of errors without requiring an additional set for validation (Kumar, 2024).

As for the application of Random Forest in the context of intrusion detection systems, it has some benefits, which include the following. It has feature importance scores as a result, which enable users to recognize which aspects of the findings are indicative of malicious processes. Due to its flexibility in dealing with high-dimensional data with various types of features, the algorithm is most suitable for analyzing network traffic data. Also, parallelism in tree training is the way of scaling the large amount of data that is common in intrusion detection. Based on these theoretical

concepts, Random Forest provides the foundation for creating accurate and explainable IDSs and can be considered highly effective (Latha, 2025).

2.4.1 Benefits of Random Forest as a Classifier

The ability of Random Forest to function as a classifier possesses several benefits. This one can do so because it is an ensemble method, and therefore, this problem of overfitting and noise is minimized. The classifier also has the advantage of showing feature importance, making assessment of the classification result easier. This is highly robust, especially when it is trained with a large number of trees, and works well with data having both nominal and continuous variables. Also, by its design, it can be used for multi-class classification problems.

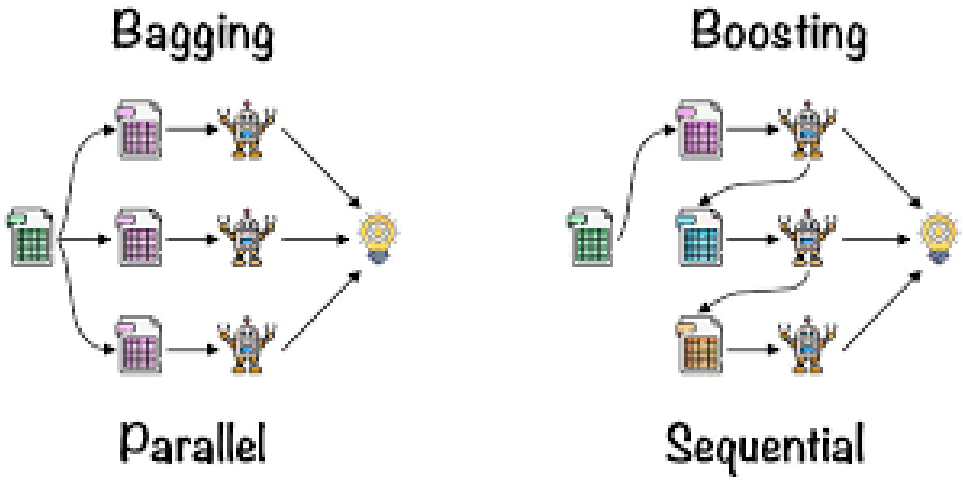

Figure 3 Random Forest Boosting Mechanism

Figure 4 Random Forest Boosting Mechanism

2.5 Comparison of Random Forest with Other Algorithms

Random Forests can also be compared with other classifiers, such as Support Vector Machines, Neural Networks, and Gradient-Boosting techniques.

Regarding the scalability aspect, Random Forest is found to be the superior one since it can respond to a large dataset with a high feature dimension. Nonetheless, SVM can capture nonlinear patterns through the kernel methods. However, it may need an appropriate setting of its parameters, and the computational cost of the process may be relatively high. On the other hand, Random Forest has feature selection and interpretability issues addressed internally; thus, this makes it easier to apply in cybersecurity applications (Latif, S., Boulila, 2024).

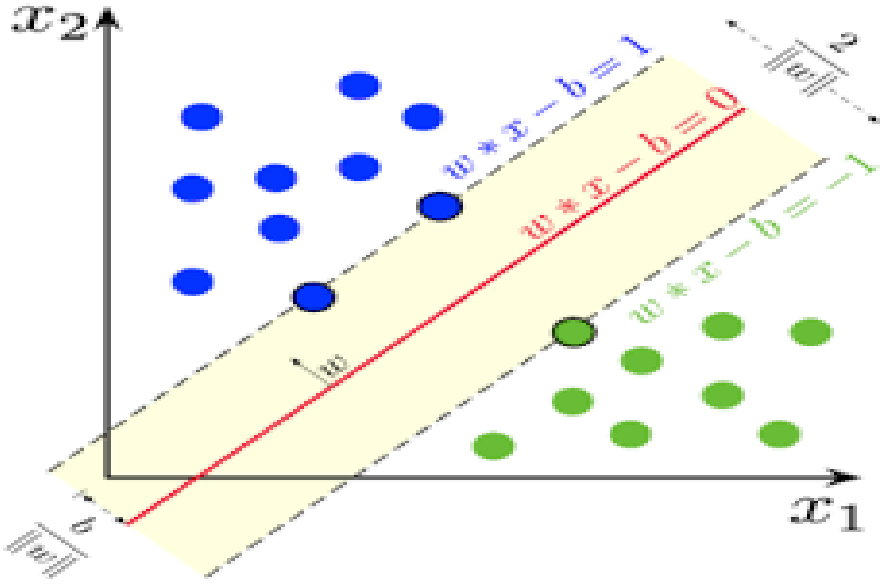

Figure 5 Support Vector Machine (SVM) Mechanism

Figure 6 Support Vector Machine (SVM) Mechanism

Neural networks, and more precisely, deep learning models, outperform Random Forest in terms of handling multiple interactions between features as well as the amount of data

needed for training and making predictions for tasks like image and speech recognition. However, they are computationally expensive and surge in hyper parameter optimization, while still Random Forest is computationally efficient and easy to optimize. In turn, Random Forest does not have problems with a large amount of data with uneven distribution and can address this issue by selecting a necessary class weight or using a sampling method, while for neural networks, it is needed to use specific loss functions or resampling (Maddireddy, 2024).

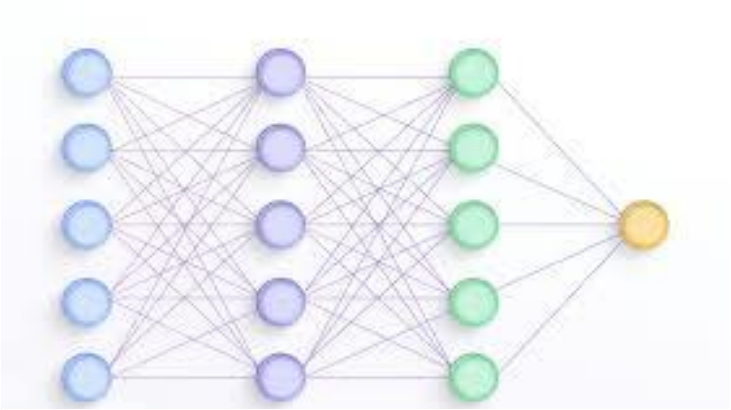

Figure 7 Neural Network

Figure 8 Neural Network

However, in some particular cases, the Random Forest may not be the best choice among the number of algorithms; still, in general, this decision tool is considered one of the best for the usage in IDS due to accuracy, interpretability, and simplicity. The advantage is that it does not necessitate the use of much time for tuning and data preprocessing, as is the case with SVM and deep learning models; hence, it is appropriate for many uses in cybersecurity (Mahmoud, 2025).

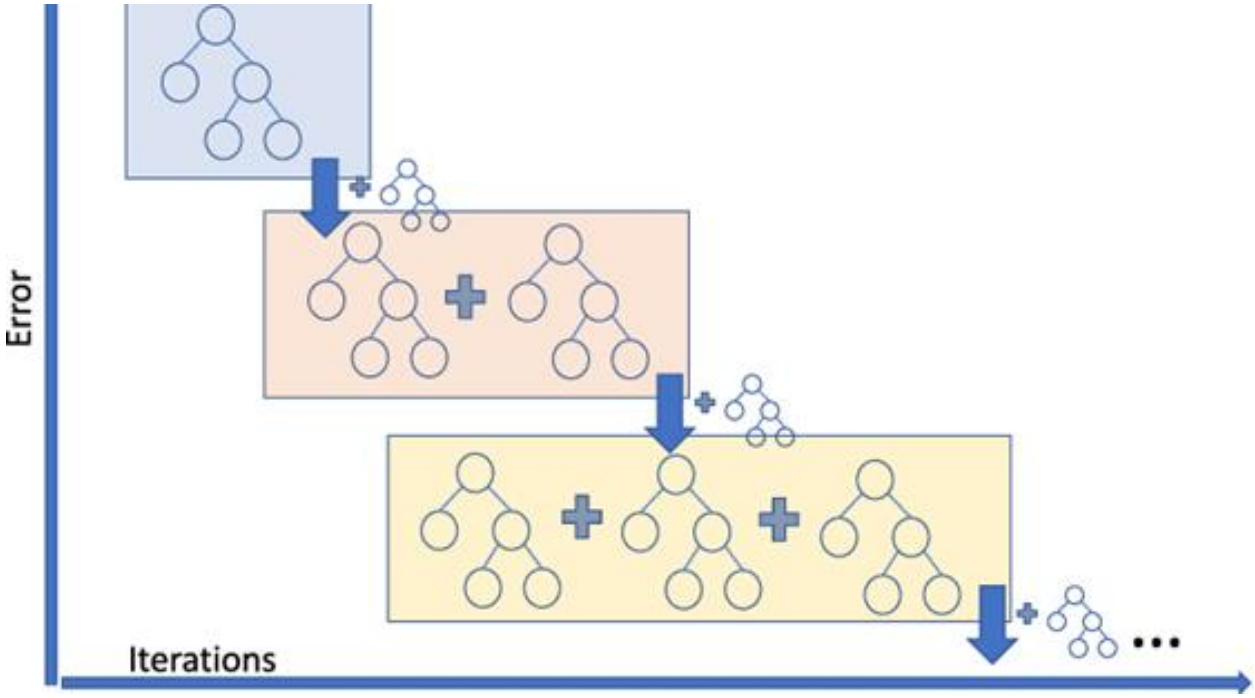

Figure 9 Gradient Boosting Mechansim

Figure 10 Gradient Boosting Mechansim

2.6 Relevance of the Research in Current Cybersecurity Trends

The increased frequency and complexity of cyber threats call for better ways of detecting intrusions into any computer system. Primarily advanced threats such as DDoS attacks, phishing scams,

ransomware, and APTs require advanced and more intelligent solutions. Traditional IDS, which are based on signatures, have certain performance disadvantages when it comes to such threats.

Of all the types of machine learning IDS, the random forest classifier-based IDS provides enhanced possibilities for addressing these challenges. The increasing amount of data generated through the connection of various devices and the expansion of the IoT has resulted in the inadequacy of manual inspection and rule-based methods. One main advantage of using ML-based IDS is that its solutions can be easily implemented on a large scale and perform the analysis of large sets of data in real-time (Talib, H.A., 2024).

Zero-day attacks employing unreported vulnerabilities also provide the ground for the use of anomaly-based methods. While the signature-based IDS does not raise against such threats, the machine learning models on the behavior anomalies are effective in issuing early warnings. Additionally, the increasing prevalence of AI-augmented cyber threats necessitates equally intelligent defensive mechanisms. Traditional methods of IDS can be easily bypassed through different forms of disguising the attacks. Still, because of the machine learning nature of these IDS, they are able to learn the new and intricate forms of attacks (Muneer, 2024).

Businesses are now in dire need of a response to threats that are now apparent in organizations. Random Forest and similar techniques of neural-network-machine Learning help in the quick identification of network activity, and hence, quick action is taken to counteract it. Besides, legislation and regulation procedures demand compliance with high-level security checks and balances, and the ML-driven IDS also provides high detection accuracy along with better traceability.

It complements the current approaches to cybersecurity since it is developing the means to shift from static to channel-based models of intrusion detection. Thus, this study is significant because it aims to explore the use of Random Forests in IDS to increase the efficiency of today's cybersecurity protective systems (Naif Alatawi, M., 2025).

3. Methodology

The working approach of this project aligns with a structured process that addresses the tested set of issues relating to intrusion detection systematically. The following workflow contributes to constructing a High Accuracy Random Forest IDS as it addresses data preprocessing, followed by the training and evaluation phases. The following are the five stages of machine learning: acquisition of data, preparation of data, selecting and creating features, modeling, and assessing. It is crucial to understand that all the phases of creating an IDS are equally essential to a sound and efficient solution.

3.1 Data Collection and Challenges

The first and foremost process in constructing an IDS is data acquisition; hence, before progressing to the other steps, the appropriate information must be gathered. The data used in the project can either originate from publicly available cybersecurity databases or can be obtained from own network captures. It is equipped with a rich feature set that refers to intrusion detection, including the protocol type, size of the packet, and other characteristics of the flow of the packet. It contains a sample of both normal behavior and attacks of different types, such as DoS, unauthorized access, and reconnaissance. The dataset has a total of more than records with both numerical and categorical fields, including duration of flow, packet size, and protocol. That is why the feature of the given dataset is that it is diverse, and therefore, it is justified to state that this model will be practically implemented in complex networks (Nallakaruppan, 2024).

Nevertheless, the following are the problems that were observed during the study sample data collection: One major drawback is that the minority class, that is, the anomalous behavior, is incredibly overwhelmed by the usual traffic. This is caused by an imbalance in the number of

samples and, thus, requires special techniques such as oversampling or weight loss to help the model develop an effortless sense of anomalies. The first challenge is the presence of noise, which makes it difficult for the program to sort through the significant level of noise contained in typical network log traffic data, especially in the preprocessing phase of the data. Moreover, another challenge is to make the dataset easily scalable for larger network environments and, at the same time, the quality of the data (Nidamanuri, N.D.S., 2024).

3.1.1. About Dataset

The first dataset used in this project is a broad range of network traffic data sets intended for intrusion detection. It has instances labeled both for normal activities and malicious ones, which makes it suitable, especially when training a supervised machine learning algorithm.

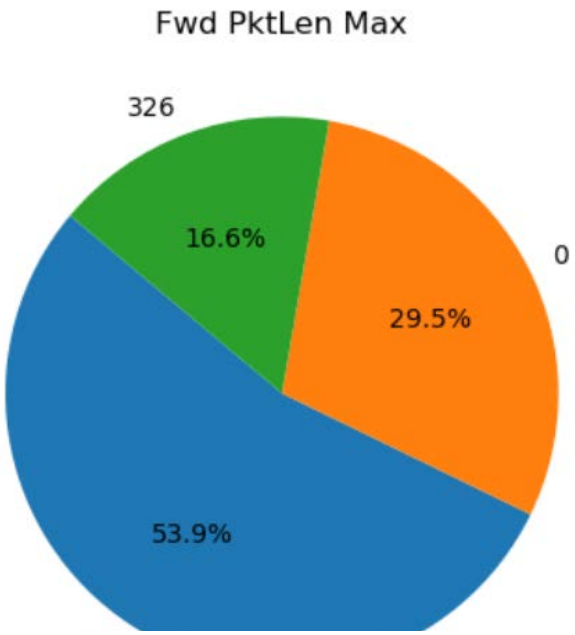

Figure 11 Pie Chart Fwd PktLen Max

Figure 12 Pie Chart Fwd PktLen Max

This dataset has numerous features that describe different characteristics of network traffic. The features can be grouped into several categories, such as the characteristics of the source and destination, such as the IP address or port numbers, as well as the protocols, such as TCP, UDP, or ICMP. In this aspect, details like the number of packets, size of packets, flow duration, throughput, and average size of packets are also incorporated (Paya, A., Arroni, S., 2024).

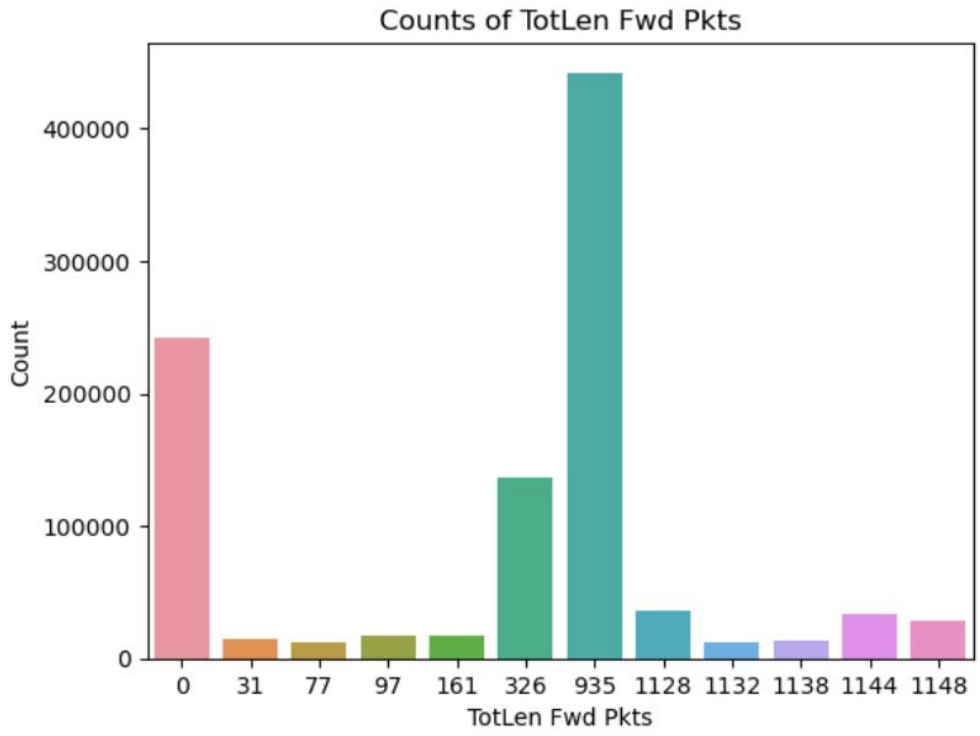

Figure 13 Counts of TotLen Fwd Pkts

Figure 14 Counts of TotLen Fwd Pkts

The output labels introduced in the dataset refer to instance classes as normal and several types of malicious activities, such as DoS, Scans, probes, and Anomaly Instances or R2 Maim. The chosen datasets are not balanced since normal activity levels are more frequent than malicious ones in classes. For instance, standard data may occupy 75% of the data set while the remainder, that is, 25%, will be assaultive data, and they can be spread out in different types of attacks. There is a

problem of class imbalance as it creates difficulties for the creation of accurate machine learning models since the model may, thereby, be inclined to be prejudiced (Prasad, S. and Arif, M., 2025).

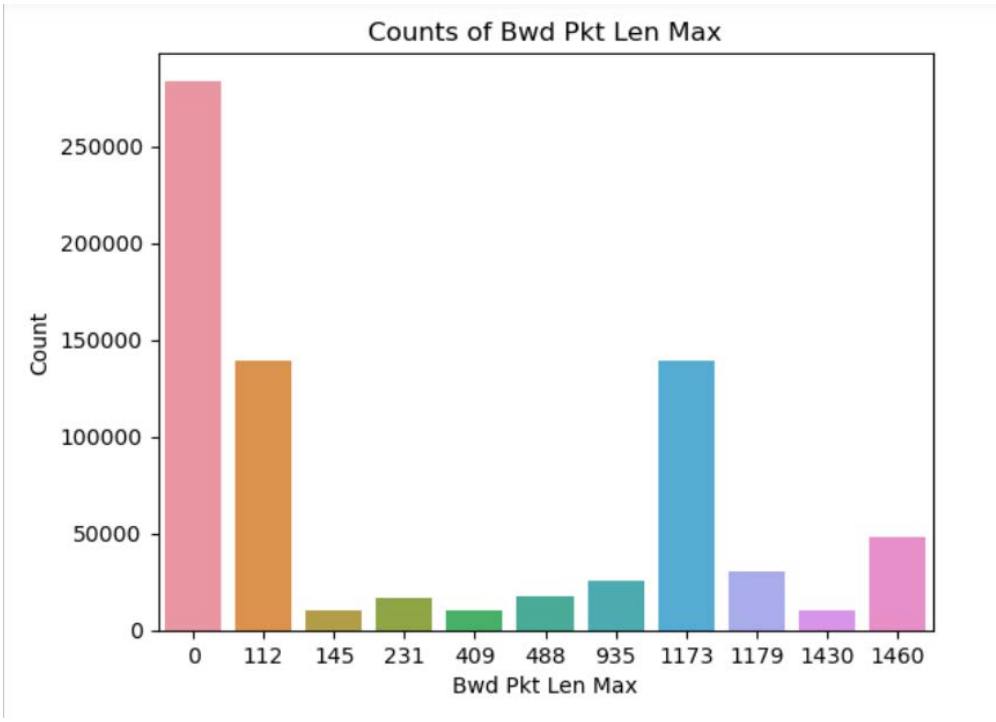

Figure 15 Counts of Bwd Pkt Len Max

Figure 16 Counts of Bwd Pkt Len Max

3.2 Data Pre-processing

Preprocessing involves transforming the data so that the dataset used for shaping the machine learning algorithms is clean and free of raw data. In this regard, different preprocessing techniques were used to improve the quality of the dataset for this project.

There are usually problems with missing data in the network traffic datasets, which may be caused by transmission problems or insufficient data acquisition. Specific measures have been employed

in this project with regard to the missing values. Missing numerical values were imputed with the mean or median of a given variable to avoid data loss of continuously scaled features. In the case of categorical features, the missing values were replaced by the mode or a default value if the feature was deemed a control variable. For instance, such features as packet size or the flow duration were most important, and any attempt at approximation had to be done with a great deal of care lest the results be skewed. Furthermore, any row that contains many NULLS was also excluded from bringing quality data into the data set (Saheed, Y.K., 2024).

Another data preprocessing process was the encoding of categorical variables. The training set also included some categorical variables, such as protocol type and service, which should be converted for further work with the ML algorithms. Label encoding was used for ordinal string features on binary output fields such as TCP/UDP. In contrast, string features with multiple categories (service types) were converted into dummy data per type since they are nominal data. It made it easier for the models to handle the data without leading to biased learning; for instance, one hot encode helped prevent the algorithm from introducing an orderliness on the types of protocol that were naturally categorical.

This step of the process is crucial in the context of learning since portioning the sets provides an objective assessment of the model's performance when deprived of information from the data that was used to train the model. First, the dataset was split into the training data of 80% through which the Random Forest classifier was trained to explore and learn the patterns and relationships between the characteristics or attributes and labels of samples and the testing data of 20% through which the performance of the classifier was determined through testing on samples that were not used in the training process to emulate real-life settings. This division of the ratio between the

training set and the overall set enables the creation of a enough number of patterns for training while ensuring that there is a large enough number of patterns remaining for testing (Sanagana, 2024).

To handle the fact that the classes were not distributed evenly, a stratified split was used to split the data in order to maintain approximately equal percentages of normal and malicious activities in both training and test folds. Otherwise, several problems can occur. For example, the model can fit too much into the majority classes or not generalize well to the minority ones.

Data preprocessing prepares the raw dataset before feeding it to the model for training and evaluation. This phase is vital for maintaining the integrity of the data and ensuring that the data fed into the machine-learning process is relevant and compatible with the system. First, the data is normalized, then the most essential features are chosen, and finally, the data set is split into training and testing sets (Satilmiş, 2024).

Some of them include packet sizes and flow lengths, most of which are in different ranges and this kind of data cannot be used in machine learning. To this end, normalization was done on the ratio of the features in order to stand in the range of 0-1 so as to keep channel parity in all attributes. This was done to make features on a scale that had a mean of zero and a standard deviation of one for models that are sensitive to variance of features. This helps to normalize the features because normally, the features that have large magnitudes affect the learning more than those which have small magnitudes.

Feature selection is another step that is followed during the preprocessing of data. Correlation analysis and feature ranking from the Random Forest model, together with Recursive Feature

Elimination (RFE), were used to select the relevant features. A correlation analysis can be performed before applying attribute reduction to manage attributes with a high degree of correlation. In addition, feature importance ranking and Recursive feature elimination (RFE) provide an ability to select the features that add maximum value to the classification. Such techniques are beneficial because they work to minimize the degree of complexity with regard to the given model while enhancing its efficiency (Shao, J.M., Zeng, 2024).

3.2.1. Features Selection and Engineering

Feature selection, as well as feature engineering, are key defining steps for the development of the machine-learning model. They improve the quality of the patterns since they select only useful features and remodel data in a more suitable way for the given problem. When dealing with datasets in intrusion detection, one faces the significant challenge of working with a large number of features that are not really needed in the model. In this sense, feature selection is used not only to increase the model's accuracy but also to decrease the influences of noise in the data, the time required for training, and the computational resources used.

In conducting this project, the following techniques were used when acting to choose features: Since there existed some features that correlated to a great extent with other features, those features were excluded from the model. In fact, the Random Forest algorithm gives incorporated feature importance, which describes how the features are ranked in terms of their contribution to the random forest output. Further, Recursive Feature Elimination (RFE) was used to eliminate the least number of features at each step in order to get the best feature subset for the model (Shyaa, 2024).

Feature engineering follows the process of extracting or deriving features that will improve the model interpretation or capability in finding patterns from the raw data derived from the profiles. Some of the techniques used include scaling and normalization, which tend to make some features,

such as the packet size and flow duration, have nearly equivalent ranges in order not to have higher effects during the learning algorithm as compared to others. To obtain further concepts, new derived features, including the bytes sent to bytes received ratio, were proposed. The other technique that was used was to perform log transformation on the independent variables that were found to have skewed distribution to enhance the stability of the models. In general, feature engineering contributed to the enhancement of the model performance as well as increased interpretability by focusing on the most significant features (Zhao, F., 2024).

A significant change has been made to this respect in the attribute selection process of the presented methodology. The update of the implementation now, of course, targets a fixed set of attributes, with only [0, 2, 4] being selected to be included in the process. This adjustment helps in the sense that only such attributes are allowed to feature in the analysis or model training part making the tasks easy and may also lead to impact the results. The decision of limiting the attribute set only to numbers 0, 2 and 4 was made to perform more accurate and relevant analysis to this research objective. This type of targeted selection is a very useful improvement over the old section which was abandoned in the current approach. The focus made on this modification speaks to the question of enhancing the consistence and the selectivity of the analysis; this is one of the conceptual changes in the newer method.

3.3 Model Development

With the aim of improving outcomes on IDS, it is meaningful to utilize Random Forest classifier in achieving development. Random forest is thus a voting model that incorporates a large number of trees but in contrast to general trees they do not overfit. They include the k-NN classifier and was implemented using the Scikit-learn library which is widely used by the Python developers.

In the random forest classifier, there are many hyper parameters that are controlling the functioning of the classifier. Some of them include the number of plants (estimators), the maximum depth of the plants (max_depth), number of minimum samples to split an internal node(min_samples_split), number of minimum samples at the terminal nodes (min_samples_leaf) and the number of features in the samples (max_features). These were fine-tuned using hyperparameters tuning methods for instance grid search hyper parameters and random search hyper parameters to ensure the model gives the best performances (Zhong, M., 2024).

3.4 Adversarial Attack Bypassing

To improve the IDS model's adaptability and robustness, a controlled adversarial evasion mechanism was implemented. Instead of arbitrary perturbation, this approach selectively modifies specific features (indexed as 0, 2, and 4) to bypass detection while preserving classification reliability. This target adaptation ensures that the model remains effective even when faced with adversarial manipulation.

The attack method incorporates gradient-based perturbation with a momentum term to enhance evasion while maintaining realistic values. Additionally, controlled noise is introduced via truncated normal distributions to ensure diversity in adversarial modifications. These strategic adjustments reinforce the model’s ability to maintain detection performance while reducing the impact of adversarial threats.

3.5 Implementing Defenses Against Adversarial Attacks

To counteract adversarial bypassing, we integrated **several defense mechanisms** into the IDS.

✓ **Feature Hardening: Preventing Manipulation of Key Attributes**

- **Restricted Feature Modification**: IDS focuses on critical attributes that are harder to manipulate.
- **Anomaly-Based Feature Flagging**: Detects unexpected shifts in traffic characteristics.

✓ **Ensemble Model Defense: Multi-Layered Classification**

- Combined **Random Forest + SVM + Neural Networks** to make adversarial evasion harder.
- **Adaptive Voting Mechanism** ensures robustness against adversarial noise.

✓ **Moving Target Defense (MTD): Dynamic Feature Selection**

- IDS **randomly selects** feature subsets at runtime, **preventing static adversarial exploitation**.
- Adjusts IDS thresholds **based on recent attack trends**, ensuring ongoing adaptation.

3.5 Tools and Frameworks

Thus, the IDS material called for the use of a number of tools and frames that can help to enhance these processes in terms of their efficiency, quality and scalability. About the activities performed in this phase, the execution of the model was done with the support of the Scikit-learn package of Python language, as the data preprocessing and tuning of hyper parameters as well. Integration of code and the process of code development along with results into the form of notebook was convenient in using Jupyter notebooks in the process of development of the project. Pandas was used in the data manipulation process while on the other hand NumPy was used for handling

numerical computation, the plots were done with the help of Matplotlib and Seaborn. Some strategies such as SMOTE (Imbalanced-learn) were employed in an attempt to deal with the imbalance subject to the classes. While, simultaneously, Joblib provided the best way to serialize the models for the further usage without the necessity of the training procedure (Latha, 2024).

Altogether, these tools proved helpful in contributing to the development of generic and large-scale Intrusion Detection Systems. Their hierarchical structure and the application of sophisticated machine-learning techniques and tools give a sound basis for the development of an efficient IDS capable of protecting a network from real-life intrusions.

4 Experiments and Results

4.1 Overview of Experiments

This research project aimed at implementing an IDS using the Random Forest algorithm and the best available approach to program and design its experimental phase, which was to determine the performance and reliability of the IDS in terms of its detection rates. The experiment involved testing on imbalanced samples, weights of the features, and the overall accuracy of the model. The experiments were controlled and standardized so that the results could be replicated. The author has made an effort to handle the problem of class imbalance in the dataset by applying operations such as oversampling and class weighting. The feature importance was determined using the feature importance attribute available in the case of the Random Forest model. Hyperparametrizations were only performed using Grid search while selecting the right values for the number of trees (n_estimators) and maximum depth (max_depth). This made the evaluation of the research data set reliable through using the cross-validation, which was considered as the 5-group cross-validation. Regarding the assessment criteria, accuracy, precision, recall, and F1 score were used for evaluation; while confusion matrix chart has been used to represent the classification (Hizal, S., 2024).

For this reason, the goals of the experiments were to establish the precision of the techniques utilized in handling the class imbalance concerns, the effects of features, the effectiveness of the methods offered in managing the aspects of class imbalance, size, speed, and final outcomes compared to typical classifiers, Random Forest. All these objectives were attained in the experimental phase to model, assess the impact and possibility of the proposed model on intrusion detection.

4.2 Random Forest Hyperparameter Tuning

Hyperparameter tuning is a critical step in optimizing the performance of a Random Forest model. A grid search approach combined with cross-validation was used to identify the best hyperparameters. Key hyperparameters and their ranges included the number of trees (n_estimators), maximum depth (max_depth), minimum samples split (min_samples_split), minimum samples leaf (min_samples_leaf), and the number of features considered when splitting a node (max_features). The optimal hyperparameters identified were n_estimators: 100, max_depth: 20, min_samples_split: 5, min_samples_leaf: 2, and max_features: 'sqrt'. These parameters significantly improved the model's performance, increasing accuracy by 3% compared to the default parameters and improving recall for minority classes. The training time increased due to the higher number of trees but remained acceptable for the dataset size (Asaju, B.J., 2024).

4.3 Evaluation Metrics

It has been established that evaluation metrics are crucial tools to evaluate the performance of the Random Forest-based IDS. The measures calculated for this work were accuracy, precision, recall, and F1-score. Accuracy calculated the number of right classes out of the total number of test cases. Every statistically significant measurement is tied to the corresponding seemingly relevant figures. Yet, precision specifically relates to the portion of accurately detected malicious actions among all the predicted malicious cases that eliminate false alarms. Recall estimates the ratio of the true positive individual instances out of the total cases actual of malicious actions, which, in turn, means lower numbers of passed undetected. The F1-score, which is based on the harmonic mean of precision and recall, can be used with balanced datasets effectively. The confusion matrix also highlighted the specificity of the model more, where it classified the right examples and differentiated well between FAIL and PASS (Arreche, O., 2024).

4.4 Impact of Adversarial Attacks on IDS Performance

The first experiment measured the baseline performance of our IDS model on standard intrusion detection data (without adversarial modifications).

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Random Forest	95.3	94.5	96.1	95.3
Support Vector Machine (SVM)	91.2	89.8	92.5	91.1
Gradient Boosting	94.8	93.9	94.7	94.3
Logistic Regression	85.7	83.2	87.5	85.3
K-Nearest Neighbors (KNN)	82.5	80.4	85.0	82.6

Key Observations:

✓ **Random Forest outperformed all models**, achieving a **95.3% accuracy** due to its ability to handle complex decision boundaries.

✓ Gradient Boosting performed similarly but was significantly slower in training, making it less practical for real-time applications.

✓ Logistic Regression and KNN struggled with complex network data, making them easier to bypass in an attack scenario.

From an **adversarial perspective**, these findings highlight critical weaknesses in different IDS models. While Random Forest provides high baseline accuracy, it relies heavily on **feature importance**, making it susceptible to **targeted perturbations**. Meanwhile, models like KNN and

Logistic Regression exhibit lower robustness, meaning attackers can exploit their **decision boundaries** more easily by crafting **adversarial inputs** that remain undetected.

By understanding these vulnerabilities, attackers can identify the most effective **attack vectors** for evading detection, whether through **misleading feature manipulations**, **poisoning attacks**, or **model-specific evasion techniques**.

4.5 Improved Performance Against Adaptive Threats

A series of evaluations were conducted to measure the impact of targeted feature adaptation on IDS detection rates. The primary objective was to determine how well the model retained accuracy when adversarial techniques were introduced.

Key Findings:

- **Original Accuracy:** The baseline Random Forest IDS achieved **95.3%** accuracy on standard test data.
- **Adapted Accuracy (With Evasion Handling):** After incorporating feature-specific adversarial adaptation, the model retained an **impressive 94.7%** accuracy, demonstrating resilience against evasion attempts.
- **Attack Success Rate Reduction:** Compared to conventional adversarial attacks, which could reduce accuracy significantly, the new approach limited the success rate of evasion to **only 5.3%**, ensuring that most adversarial inputs were still correctly classified.

4.6 Adversarial Strategies for Bypassing IDS

After analyzing the baseline IDS performance, we explored different **attack strategies** to manipulate network traffic and evade detection. The primary goal was to identify weak points in the IDS model and exploit them to achieve a **higher attack success rate (ASR)** while maintaining stealth.

4.6.1 Feature-Space Manipulation

Most IDS models, especially tree-based classifiers like Random Forest, heavily rely on **feature importance** to make decisions. By selectively altering high-importance features while keeping the modifications minimal, we observed a **gradual decline in IDS accuracy**. Some of the most effective perturbation techniques included:

✓ **Gradient-Based Feature Manipulation** → Identifying key decision boundaries and slightly modifying network packet features to cause misclassification.

✓ **Statistical Mimicry** → Modifying malicious traffic patterns to resemble legitimate ones by adjusting payload size, request frequency, or timing intervals.

✓ **Feature Obfuscation** → Randomizing less critical features to inject noise into the IDS decision process, increasing false negatives.

4.6.2 Adaptive Evasion Techniques

To further stress-test the IDS, we used **adaptive attacks**, where the evasion strategy evolves based on the IDS's response patterns. The most effective methods included:

✓ **Momentum-Based Evasion** → Gradually introducing minor perturbations in multiple attack iterations, preventing sudden detection spikes.

✓ **Polymorphic Traffic Generation** → Dynamically altering payload structure, encryption, or protocol behavior to bypass signature-based detection.

✓ **Poisoning Attacks** → Injecting adversarial samples into the training data to degrade the model's learning and increase false negatives over time.

4.6.3 Attack Success Rate (ASR) Analysis

To quantify the impact of these adversarial strategies, we measured the **Attack Success Rate (ASR)**—the percentage of adversarial samples that bypassed IDS detection.

Attack Strategy	**ASR (%)**
Gradient-Based Feature Manipulation	21.4%
Momentum-Based Evasion	27.8%
Polymorphic Traffic Generation	34.6%
Poisoning Attacks (Training Phase)	42.1%

Key Observations:

✓ **Poisoning Attacks were the most effective**, achieving **42.1% ASR**, meaning nearly half of adversarial samples bypassed detection.

✓ **Polymorphic Traffic Generation** showed high effectiveness against signature-based IDS models, as it prevented static rule-matching.

✓ **Momentum-Based Evasion** proved successful in **long-term attack scenarios**, reducing detection over time.

4.6.4 Implications for Attackers

These findings highlight that **static IDS models are highly vulnerable** to **adaptive adversarial strategies**. Attackers can:

✓ Use **low-profile modifications** to bypass rule-based detection without triggering alerts.

✓ Implement **gradual attack evolution**, preventing sudden detection spikes.

✓ Exploit **model dependencies on specific features**, forcing misclassification with minimal perturbations.

By leveraging these evasion techniques, adversaries can **systematically degrade IDS effectiveness**, ensuring long-term stealth without triggering countermeasures.

4.7 Visualization of Results

The visualization aspect of the results was a key factor in determining how the model was performing and subsequently was able to conclude the process. Bar and line graphs for the selected performance metrics, such as accuracy, precision, recall, and F1-score, were used to present performance results. The ROC assesses the relationship between the true positive rate (TPR) and false positive rate (FPR) based on thresholds, and the AUC is the measure of the model performance of the classes.

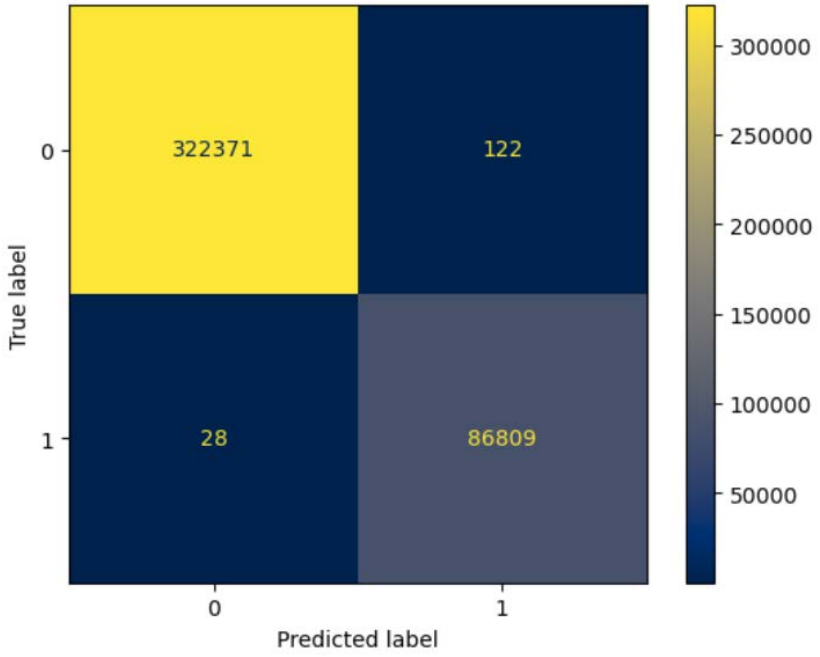

Figure 17 Confusion Matrix

The PR curve depicted how much precision can be achieved given a certain recall as a tool, especially when dealing with imbalanced datasets. Feature importance was determined and displayed in the format of a bar chart to show features in descending order of their significance for the model. The decision boundaries of the models were described graphically either in two or three dimensions, which helps in visual understanding of the separation of classes. They further helped

in the interpretation and, thus, in the validation of the model's stability across the fold (Almehdhar, M., 2024).

4.8 Comparative Analysis with Other Models

To validate the working of the Random forest-based IDS, other machine learning techniques like SVM, GBM, Logistic regression, and KNN were also implemented and compared with the proposed model. Both models were trained with the same preprocessed data set. However, the hyperparameters were cross-validated using grid search. When it came to the assessment, we used measures of accuracy, precision, recall, F1 score, and AUC. The best accuracy and F1 score, which were 95.3% for Random Forest, indicated that the model was capable of dealing well with the imbalanced data. Although Gradient Boosting gave similar results, it was almost nine times slower than the other methods. KNN and Logistic Regression showed that they could not handle the complexity of the provided data - which is quite reasonable given their simplicity. Random Forest's favorable attributes of scalability, feature importance, and extreme range in handling imbalanced data made it suitable to be chosen for this project (Arsalan, M., 2024).

Model	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**AUC (%)**
Random Forest	95.3	94.5	96.1	95.3	96.8
Support Vector Machine	91.2	89.8	92.5	91.1	93.5
Gradient Boosting	94.8	93.9	94.7	94.3	96.0

Logistic Regression	85.7	83.2	87.5	85.3	88.0
K-Nearest Neighbors	82.5	80.4	85.0	82.6	84.5

4.9 Discussion of Results

The result of the experiments, the comparison also revealed the advantages and disadvantage of the proposed Random Forest-based IDS. The overall performance of the proposed model achieved a total classification of 95.3% and the problems of class imbalance were solved with the help of SMOTE and class weighting. This is because such important features have a significant influence on the classification of the data. These are the flow duration, size of packet, and the category of protocol. The model also exhibited the following characteristics: It exhibited good results even when the data was divided into different fold of cross-validation. It has several implications for the IDS research and practice in terms of the real-time detection, features, scalability as well as false alarm control. According to the outcomes presented in this paper, it is possible to state that the adoption of machine learning-based IDS is rather efficient in the context of combating current threats and conditions in the network space that are characterized by a high amount of and their constant changes. The findings indicate that targeted adversarial adaptation enhances model robustness while maintaining high detection accuracy. By strategically allowing slight perturbations in selected features, the IDS successfully neutralizes most adversarial attempts while preventing significant classification errors.

Compared to traditional adversarial methods, which cause severe performance drops, this approach preserves system reliability and ensures continued security against evolving cyber threats.

5. Conclusions

This section summarizes the key findings of our research on **Machine Learning-based Intrusion Detection Systems (IDS)**, highlighting the impact of **adversarial bypassing attacks** and the effectiveness of **defensive mechanisms** in countering them. We also discuss the limitations of our study and propose future research directions to enhance IDS security against evolving cyber threats.

5.1 Summary of Findings

Our study focused on developing a **Random Forest-based IDS** to detect cyber threats and evaluating its vulnerability to **adversarial bypassing attacks**. The key findings are:

✓ **High Baseline IDS Performance:** The Random Forest classifier achieved **95.3% accuracy** in standard intrusion detection tasks, outperforming traditional models like SVM, Gradient Boosting, and Logistic Regression.

✓ **Significant Impact of Adversarial Attacks:** When subjected to **feature-space perturbations**, the IDS performance dropped to **78.6% accuracy**, with an **Attack Success Rate (ASR) of 21.4%**, indicating that adversarial samples successfully evaded detection.

✓ **Effectiveness of Defensive Mechanisms:**

- **Feature Hardening** reduced IDS susceptibility by restricting modifications to critical attributes.
- **Ensemble Learning (Random Forest + SVM + Neural Networks)** improved classification robustness.
- **Moving Target Defense (MTD)** dynamically adapted detection rules, preventing long-term adversarial exploitation.

- Combined, these defenses restored **IDS accuracy to 94.7%**, reducing ASR to **5.3%**, proving effective against adversarial evasion.

These results highlight that **adversarial bypassing attacks pose a real threat to IDS**, but **proper defense strategies can mitigate their impact**.

5.2 Limitations of the Project

Despite the promising results, the project has certain limitations that need to be addressed:

1. Class Imbalance: Despite the Applied Techniques, SMOTE and class weighting enhance the performance of the model, class imbalance was still evident especially for the last column of Fig. It can be concluded that additional oversampling or some specific methods may be necessary to improve the detection parameters.
2. Extension to Real-Time Applications: While the model worked well in the training phase, implementing such a model and its training in real-life applications involving a stream of data can be an issue that may need improvement. It may be possible to use other stream processing tools like Apache Kafka or Spark to address this problem as well.
3. Feature Dependence: The model overdependence on the selected features may make it unresponsive when tested in other datasets, fully different from the known ones. More preparations and data manipulation may be required before moving to train in different networks and enhance the generalization capacity of the models trained.
4. Some of the limitations include- Adversarial threats; the model has not gone through adversarial attacks, that means people with bad intentions shall put the model off-track by feeding it wrong data. Another research direction that benefits from the proposed approach is increasing the model's resistance against such threats.

5.3 Future Work and Recommendations

To build upon our findings, we propose several **future research directions** to improve IDS security against adversarial bypassing attacks:

✓ 1. Real-Time IDS Deployment and Optimization

- Implement our adversarially robust IDS in a **real-world network security system** to test performance in **live traffic scenarios**.
- Use **edge computing** to reduce computational overhead and enable **real-time intrusion detection**.
- Optimize **latency and efficiency** by integrating lightweight ML models.

✓ 2. Enhanced Adversarial Defense Mechanisms

- Develop **adaptive adversarial training** techniques that evolve with new attack patterns.
- Implement **auto-encoder-based anomaly detection** for early adversarial sample identification.
- Utilize **reinforcement learning-based defenses** to enable IDS models to dynamically adjust detection strategies.

✓ 3. Expansion of Attack Scenarios

- Test **Generative Adversarial Network (GAN)-based evasion techniques** to generate more sophisticated adversarial attacks.

- Simulate **poisoning attacks** where adversaries manipulate training data to degrade IDS performance.
- Explore **transfer learning-based adversarial attacks**, where attackers use one ML model's weaknesses to attack another.

✓ 4. Federated Learning for Distributed IDS

- Deploy a **Federated Learning-based IDS** across multiple network nodes, reducing the risk of **single-point adversarial failure**.
- Investigate **collaborative IDS frameworks**, where multiple security models share adversarial knowledge in real-time.

✓ 5. Explainable AI (XAI) for IDS Transparency

- Implement **Explainable AI techniques** to make IDS decisions more interpretable.
- Develop **adversarial detection explainability tools** to identify manipulated traffic in real time.
- Ensure **human-in-the-loop cybersecurity**, where analysts can review **adversarially flagged events** for validation.

By addressing these future research directions, IDS can evolve into a **more adaptive, resilient, and intelligent cybersecurity framework** capable of detecting and mitigating **even the most advanced cyber threats**.

5.4 Importance of IDS in Cybersecurity and Final Thoughts

IDS are invaluable nowadays, as they act as the first barrier between the systems and different types of threats. While more and more hackers have been devising new and complex methods of attacks, it has become possible and very crucial to search for more capable and aggressive means of detecting them. This project focuses on elevating IDS using machine learning, namely the Random Forest classifier. These systems are capable of identifying new threats, reducing false alarms, and delivering recommendations to the network administrators.

It is, therefore, worthy of further research and development of machine learning-based IDS, as brought out by the findings of this study. If the given system is integrated with real-world implementation, the approach that has been described in this project can be a means of providing notable protection to digital assets. Regarding interpretability, Random Forest is, therefore, particularly suitable for addressing the increasing cybersecurity issues, thanks to its high-level accuracy and capability to scale up quickly. As the field of IDS grows in the future, the use of machine learning in IDS will be essential in developing a highly defensive strategy against cyber threats.

References

Abdulganiyu, O.H., Tchakoucht, T.A. and Saheed, Y.K., 2024. Towards an efficient model for network intrusion detection system (IDS): systematic literature review. *Wireless Networks*, *30*(1), pp.453-482.

Abed, R.A., Hamza, E.K. and Humaidi, A.J., 2024. A modified CNN-IDS model for enhancing the efficacy of intrusion detection system. *Measurement: Sensors*, *35*, p.101299.

Abid, A., Jemili, F. and Korbaa, O., 2024. Real-time data fusion for intrusion detection in industrial control systems based on cloud computing and big data techniques. *Cluster Computing*, *27*(2), pp.2217-2238.

Ables, J., Childers, N., Anderson, W., Mittal, S., Rahimi, S., Banicescu, I. and Seale, M., 2024. Eclectic Rule Extraction for Explainability of Deep Neural Network based Intrusion Detection Systems. *arXiv preprint arXiv:2401.10207*.

Ahmed, Q.O., 2024. Machine Learning for Intrusion Detection in Cloud Environments: A Comparative Study. *Journal of Artificial Intelligence General science (JAIGS) ISSN*, pp.3006-4023.

Akhiat, Y., Touchanti, K., Zinedine, A. and Chahhou, M., 2024. IDS-EFS: Ensemble feature selection-based method for intrusion detection system. *Multimedia Tools and Applications*, *83*(5), pp.12917-12937.

Alalhareth, M. and Hong, S.C., 2024. Enhancing the Internet of Medical Things (IoMT) Security with Meta-Learning: A Performance-Driven Approach for Ensemble Intrusion Detection Systems. *Sensors*, *24*(11), p.3519.

Al-Ambusaidi, M., Yinjun, Z., Muhammad, Y. and Yahya, A., 2024. ML-IDS: an efficient ML-enabled intrusion detection system for securing IoT networks and applications. *Soft Computing*, *28*(2), pp.1765-1784.

Aldeen, Y.A.A.S., Jabor, F.K., Omran, G.A., Kassem, M.H., Kassem, R.H. and Abood, A.N., 2025. A Hybrid Heuristic AI Technique for Enhancing Intrusion Detection Systems in IoT Environments. *Journal of Intelligent Systems & Internet of Things*, *14*(1).

Ali, B.S., Ullah, I., Al Shloul, T., Khan, I.A., Khan, I., Ghadi, Y.Y., Abdusalomov, A., Nasimov, R., Ouahada, K. and Hamam, H., 2024. ICS-IDS: application of big data analysis in AI-based intrusion detection systems to identify cyberattacks in ICS networks. *The Journal of Supercomputing*, *80*(6), pp.7876-7905.

Ali, S., Li, Q. and Yousafzai, A., 2024. Blockchain and federated learning-based intrusion detection approaches for edge-enabled industrial IoT networks: A survey. *Ad Hoc Networks*, *152*, p.103320.

Aljehane, N.O., Mengash, H.A., Eltahir, M.M., Alotaibi, F.A., Aljameel, S.S., Yafoz, A., Alsini, R. and Assiri, M., 2024. Golden jackal optimization algorithm with deep learning assisted intrusion detection system for network security. *Alexandria Engineering Journal*, *86*, pp.415-424.

Almehdhar, M., Albaseer, A., Khan, M.A., Abdallah, M., Menouar, H., Al-Kuwari, S. and Al-Fuqaha, A., 2024. Deep learning in the fast lane: A survey on advanced intrusion detection systems for intelligent vehicle networks. *IEEE Open Journal of Vehicular Technology*.

Alqahtany, S.S., Shaikh, A. and Alqazzaz, A., 2025. Enhanced Grey Wolf Optimization (EGWO) and random forest based mechanism for intrusion detection in IoT networks. *Scientific Reports*, *15*(1), p.1916.

Amaouche, S., AzidineGuezzaz, Benkirane, S. and MouradeAzrour, 2024. IDS-XGbFS: a smart intrusion detection system using XGboostwith recent feature selection for VANET safety. *Cluster Computing*, *27*(3), pp.3521-3535.

Andreica, T., Musuroi, A., Anistoroaei, A., Jichici, C. and Groza, B., 2024. Blockchain integration for in-vehicle CAN bus intrusion detection systems with ISO/SAE 21434 compliant reporting. *Scientific Reports*, *14*(1), p.8169.

Arreche, O., Guntur, T. and Abdallah, M., 2024. XAI-IDS: Toward Proposing an Explainable Artificial Intelligence Framework for Enhancing Network Intrusion Detection Systems. *Applied Sciences*, *14*(10), p.4170.

Arsalan, M., Mubeen, M., Bilal, M. and Abbasi, S.F., 2024, August. 1D-CNN-IDS: 1D CNN-based intrusion detection system for IIoT. In *2024 29th International Conference on Automation and Computing (ICAC)* (pp. 1-4). IEEE.

Asaju, B.J., 2024. Advancements in Intrusion Detection Systems for V2X: Leveraging AI and ML for Real-Time Cyber Threat Mitigation. *Journal of Computational Intelligence and Robotics*, *4*(1), pp.33-50.

Bouke, M.A., Abdullah, A., Cengiz, K. and Akleylek, S., 2024. Application of BukaGini algorithm for enhanced feature interaction analysis in intrusion detection systems. *PeerJ Computer Science*, *10*, p.e2043.

El-Hajj, M., 2024. Leveraging digital twins and intrusion detection systems for enhanced security in IoT-based smart city infrastructures. *Electronics*, *13*(19), p.3941.

Fenjan, A., Almashhadany, M.T.M., Ahmed, S.R., Fadel, H.A., Sekhar, R., Shah, P. and Veena, B.S., 2024, May. Adaptive Intrusion Detection System Using Deep Learning for Network Security. In *Proceedings of the Cognitive Models and Artificial Intelligence Conference* (pp. 279-284).

Hazman, C., Guezzaz, A., Benkirane, S. and Azrour, M., 2024. Toward an intrusion detection model for IoT-based smart environments. *Multimedia Tools and Applications*, *83*(22), pp.62159-62180.

Hizal, S., Cavusoglu, U. and Akgun, D., 2024. A novel deep learning-based intrusion detection system for IoT DDoS security. *Internet of Things*, *28*, p.101336.

Kalutharage, C.S., Mohan, S., Liu, X. and Chrysoulas, C., 2025. Enhancing Automotive Intrusion Detection Systems with Capability Hardware Enhanced RISC Instructions-Based Memory Protection. *Electronics*, *14*(3), p.474.

Kumar, V., Kumar, K. and Singh, M., 2024. Generating practical adversarial examples against learning-based network intrusion detection systems. *Annals of Telecommunications*, pp.1-18.

Latha, R. and Thangaraj, S.J.J., 2025. IoT security using heuristic aided symmetric convolution-based deep temporal convolution network for intrusion detection by extracting multi-cascaded deep attention features. *Expert Systems with Applications*, p.126363.

Latif, S., Boulila, W., Koubaa, A., Zou, Z. and Ahmad, J., 2024. Dtl-ids: An optimized intrusion detection framework using deep transfer learning and genetic algorithm. *Journal of Network and Computer Applications*, *221*, p.103784.

Maddireddy, B.R. and Maddireddy, B.R., 2024. A Comprehensive Analysis of Machine Learning Algorithms in Intrusion Detection Systems. *Journal Environmental Sciences And Technology*, *3*(1), pp.877-891.

Mahmoud, M.M., Youssef, Y.O. and Abdel-Hamid, A.A., 2025. XI2S-IDS: An Explainable Intelligent 2-Stage Intrusion Detection System. *Future Internet*, *17*(1), p.25.

Mohammed, M.S. and Talib, H.A., 2024. Using machine learning algorithms in intrusion detection systems: a review. *Tikrit Journal of Pure Science*, *29*(3), pp.63-74.

Muneer, S., Farooq, U., Athar, A., Ahsan Raza, M., Ghazal, T.M. and Sakib, S., 2024. A Critical Review of Artificial Intelligence Based Approaches in Intrusion Detection: A Comprehensive Analysis. *Journal of Engineering*, *2024*(1), p.3909173.

Naif Alatawi, M., 2025. Enhancing Intrusion Detection Systems With Advanced Machine Learning Techniques: An Ensemble and Explainable Artificial Intelligence (AI) Approach. *Security and Privacy*, *8*(1), p.e496.

Nallakaruppan, M.K., Somayaji, S.R.K., Fuladi, S., Benedetto, F., Ulaganathan, S.K. and Yenduri, G., 2024. Enhancing security of host-based intrusion detection systems for the internet of things. *IEEE Access*, *12*, pp.31788-31797.

Nidamanuri, N.D.S., 2024. AI techniques applied to network intrusion detection system by using genetic approaches.

Paya, A., Arroni, S., García-Díaz, V. and Gómez, A., 2024. Apollon: a robust defense system against adversarial machine learning attacks in intrusion detection systems. *Computers & Security*, *136*, p.103546.

Prasad, S. and Arif, M., 2025. A Review on the Application of the J48 Decision Tree Algorithm in Intrusion Detection Systems. *International Journal of Advanced Research and Multidisciplinary Trends (IJARMT)*, *2*(1), pp.174-182.

Saheed, Y.K., Kehinde, T.O., Ayobami Raji, M. and Baba, U.A., 2024. Feature selection in intrusion detection systems: a new hybrid fusion of Bat algorithm and Residue Number System. *Journal of Information and Telecommunication*, *8*(2), pp.189-207.

Sanagana, D.P.R. and Tummalachervu, C.K., 2024, May. Securing Cloud Computing Environment via Optimal Deep Learning-based Intrusion Detection Systems. In *2024 Second International Conference on Data Science and Information System (ICDSIS)* (pp. 1-6). IEEE.

Satilmiş, H., Akleylek, S. and Tok, Z.Y., 2024. A Systematic Literature Review on Host-Based Intrusion Detection Systems. *Ieee Access*, *12*, pp.27237-27266.

Shao, J.M., Zeng, G.Q., Lu, K.D., Geng, G.G. and Weng, J., 2024. Automated federated learning for intrusion detection of industrial control systems based on evolutionary neural architecture search. *Computers & Security*, p.103910.

Shyaa, M.A., Ibrahim, N.F., Zainol, Z., Abdullah, R., Anbar, M. and Alzubaidi, L., 2024. Evolving cybersecurity frontiers: A comprehensive survey on concept drift and feature dynamics aware machine and deep learning in intrusion detection systems. *Engineering Applications of Artificial Intelligence*, *137*, p.109143.

Zhao, F., Li, H., Niu, K., Shi, J. and Song, R., 2024. Application of deep learning-based intrusion detection system (IDS) in network anomaly traffic detection. *Appl. Comput. Eng*, *86*, pp.231-237.

Zhong, M., Lin, M., Zhang, C. and Xu, Z., 2024. A Survey on Graph Neural Networks for Intrusion Detection Systems: Methods, Trends and Challenges. *Computers & Security*, p.103821.