VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

Razi Sohail

CLOUD ENHANCED INTRUSION DETECTION

Evaluating Deep Learning and Ensemble Methods Using Hikari 2021 Dataset.

School of Technology

2024

VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES
Cloud-Based Software Engineering

ABSTRACT

Author	Razi Sohail
Title	Cloud-Enhanced Intrusion Detection Evaluating Deep Learning and Ensemble Methods Using HIKARI 2021 Dataset.
Year	2024
Language	English
Pages	64
Name of Supervisor	Jukka Matila

The thesis provides an in-depth analysis into machine learning and deep learning system methodologies for network invasion discovery using the HIKARI-2021 dataset. This involved the application of models like Random Forest, XG Boost, LSTM, and GRU in identifying and classifying various malicious undertakings transmitted within the network exchange.

Depending on the operation, the models were accessed primarily with reference to their accuracies and their confusion matrix evaluations. The results of this study indicated Random Forest with an accuracy of 93.77%, XG Boost at 93.02%, LSTM at 92.48%, and GRU at 92.50%. Subsequently, this was related to benchmark system models as demonstrated in related literature contrast with random performances of between 98% and 99% accuracy.

This research concludes that, based on form of entry employed, the model is imminently viable because each has its power, weakness, and potential in real-earth usage. Additionally, there is an opportunity for optimizing these system models and designing new features based on the lessons learned from this analysis.

Keywords HIKARI-2021 dataset, machine learning, deep learning, Random Forest, XG Boost, LSTM, GRU.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

1.1 Background and Context

In circumstances where nearly all spheres of human endeavor have moved online, the safety of digital networks has acquired unprecedented importance. While the internet is the primary source of communication and general commerce and entertainment in modern times, cyber threats have taken on distinctively new dimensions and complexity levels. As a result, a critical need for progressive forms of defense strategies has emerged, and network intrusion is of the most significance.

1.1.1 Development of Network Intrusion Detection Systems

The history of Network Intrusion Detection Systems is almost as old as computing itself. In the early days, all networks tended to be much smaller, more isolated, and predominantly unrelated. As a result, security threats were much more simplistic as well. Basic tools and logs were typically sufficient for admins to observe, spot, and quarantine any anomalies in network. But as networks continued to increase, connect, and expand, it became clear that maintaining security in such an environment required a more proactive approach.

The first tools that can be attributed to the modern definition of NIDS appeared in the early 1990s. These were the so-called signature-based systems. They operated on the principle of matching predefined patterns, or signatures, with known malicious activities. However, the number of threats continued to grow, so the corresponding volume of signatures became excessive, and systems failed to detect entirely new and unknown threats. Moreover, indexing the database containing signatures of all possible threats became an overwhelming task in and of itself.

In the late 90s and early ’00s, the security landscape essentially changed. Attacking parties learned to use custom malware and more advanced techniques to breach

defenses. At all times, the response systems were adjusted – this was the starting point of anomaly-based detection, in which network traffic is monitored for unexpected patterns or behavior paradigms and is marked for additional inquiry.

But even these adaptations were not enough – false positive rates for early warning systems remained high. The cybersecurity sector was in urgent need of a more accurate, adaptable, and scalable solution.

1.1.2 Contribution of Machine Learning and Deep Learning in NIDS

Machine Learning and Deep Learning found their way into the network security domain. These new technologies offered the promise of increasing current intrusion detection systems accuracy and efficiency and making these systems more flexible to the constantly evolving cyber threat scenario.

Machine learning is a subset of artificial intelligence that uses algorithms to identify patterns or regularities in any given data. In the case of NIDS, ML algorithms can use large datasets of historical network traffic to learn what ‘normal’ data look like. After training, ML models can compare new data to those learned patterns. Any substantial difference is identified as a deviation by the model which is likely to be an intrusion. ML algorithms like Random Forest and XG Boost gained popularity in other sectors and online detection. These models had their unique set of advantages, like an ability to handle large datasets and produce interpretable results. Hence, many researchers and digital security experts use them in practice.

On the other hand, deep learning, a type of machine learning, uses neural networks consisting of numerous layers. Particularly, deep learning uses neural networks with numerous layers and extensively analyzes databases. Deep learning models, involving recurrent neural networks like LSTM and GRU, have been instrumental in recognizing patterns in complex sequences in data. It enabled deep learning-based models to process network traffic files. Moreover, these

algorithms became very beneficial in a sequence-time investigation because there is a great way to store their periods.

The integration of ML and DL within NIDS provided various benefits. Primarily, the systems became more accurate in zeroing down on actual anomalies, thus reducing the rate of false positives. Secondly, they provided a coverage mechanism, often proving capable of detecting and preventing zero-day threats due to their proactive nature. Lastly, the ability of the model to continue learning ensures that they adapt to the changing landscape of the threat, creating the protection techniques more coherent over time.

1.2 Statement of the Problem

The current state of network security is advancing but challenged at the between sides also. Network administrators and cyber attackers are utilizing technological innovation and perseverance broadly against each other. On the one hand, modern Network Intrusion Detection Systems have clear advantages, but there is also a list of their aspects of high concern.

1.2.1 Present Obstacles in Network Intrusion Detection Systems

First of all, Data traffic is among the biggest problems. The Internet of Things has enabled more and more devices to be connected in use and data transmission. That said, there is a tremendous amount of data which necessitates an NIDS to be able to go through all data generated in real time which is not possible with the existing ones.

Secondly, the effective nature of cyber threats is a major issue. New forms of malware, viruses, and hacking methods are discovered daily. Signature-based NIDS are not suitable for this new trend. They are reactive since they can only identify known threats and are frequently helpless against unidentified ones.

One more issue that has been relevant for some time is false positives. Although it is essential to identify potential threats, frequent false alarms can lead to alert fatigue. In unmanageable numbers, network administrators may begin to ignore these reports, increasing the chances of ignoring real dangers. It is, in most cases, at risk of having a substantial number of false positives, which makes it more of a nuisance than a help.

Finally, the more common use of encrypted data makes the task twice as hard. While excellent for privacy and data security, encryption is a double-edged sword in the context of NIDS. Indeed, conventional detection mechanisms often are capable of decrypting and analyzing encrypted traffic, which means that it can overlook the malicious activity hiding within.

1.2.2 Integrated Techniques and Their Capabilities

Given the issues discussed above, hybrid systems where combine various methods strengths seem to be of significant interest. Hybrid systems, comprising both the machine learning and deep learning technologies, present a new boundary in the field of NIDS.

The reason is succinctly summarized as follows: while many machine learning and XG Boost Random Forest models are rightly flawless with large datasets and accurate results but might miss all intricate observances within the data. At the same time, various deep learning, such as LSTM and GRU models can grasp complex alignments and interconnection in the data that traditional machine learning base models might miss. Therefore, while combating these systems using both models at the same time, what can be achieved is both breadth and depth, which is seemingly impossible to achieve on a standalone basis.

Moreover, hybrid systems can diminish the number of false positives. By applying the machine learning role in segregating the apparent normal life and the deep

learning base model observing and coming up with intricate design, raising a fake terrible alarm is limited.

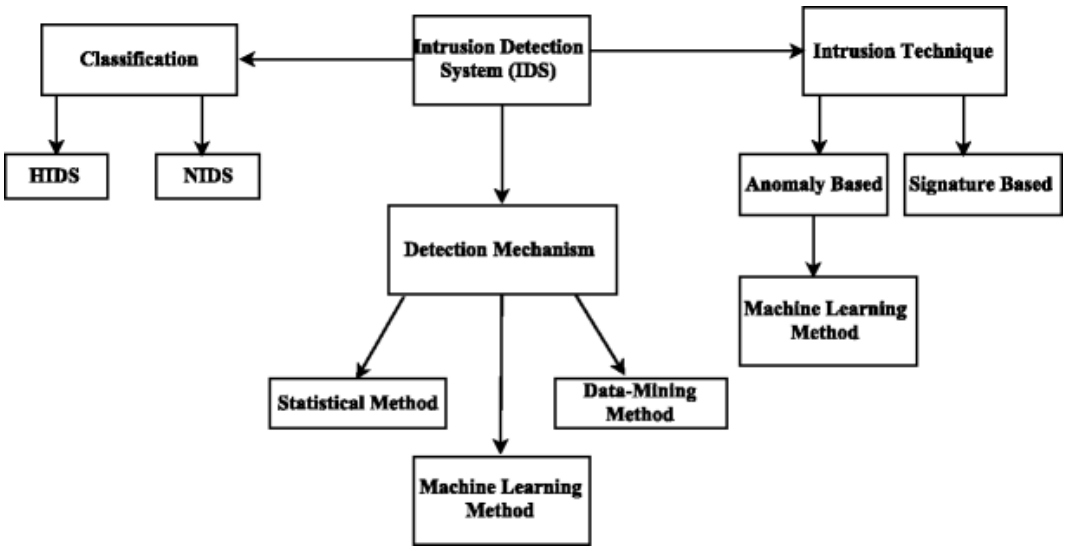

Figure 1. Overview of Intrusion Detection Systems

Flexibility is another important benefit of hybrid systems. Because ML and DL models work unattended and continuously "learn" on currently processed data, they can be trained on the fly about any new dangerous threats or about evolving network behavior. They become predictive, not just reactive, able to forecast potentially dangerous events based on previously detected patterns, which conventional signature-based systems cannot do.

1.3 Objectives of the Study

The domain of network security is huge and calls for continued exploration and innovation. The aim of the given research is to explore the integration of machine learning and deep learning in Network Intrusion Detection Systems. In particular, the objective is as follows:

1 To investigate the current NIDS solutions and point out the strengths and weaknesses of the existing systems.

2 To analyse two key contributions of machine learning including Random Forest and XG Boost algorithms as intrusion detection methods to determine their performance, apply in a wide range of use cases, and effectiveness in the real world.

3 To analyse the contribution of two deep learning algorithms LSTM and GRU layers as recognition engines of complex networking patterns to determine their efficiency in NIDS.

4 To design a hybrid NIDS system correlating the advantages of ML and DL to ensure compliances coverage and effectiveness.

5 To measure the efficiency of the model, comparing it with a traditional NIDS system to determine its performance, including accuracy, false positive rate, and ability to catch up with new cyber threats.

1.4 Research Questions

While undertaking this research, the following questions form the basis of our inquiries. The questions form the backbone of the study and deploy several forms of observation. They include:

1 How well do the current NIDS perform, considering the rapidly shifting landscape of online threats?

2 How do the machine learning algorithms, that is XG-Boost and Random-forest, contribute to NIDS, and do they provide a strong performance against real-world network attacks?

3 How effective are deep learning models, such as GRU and LSTM, in improving NIDS detection, and counter the challenge of complex network parameters?

4 Is the combination of machine learning and deep learning models a good high-positivity solution to NIDS as opposed to the conventional systems?

5 How well does the hybrid solution perform in terms of accuracy, falser under these parameters compared to existing common hybrid systems?

These questions will help to drive the research and build broader knowledge on the topic of network security. By determining the parameters and scope of machine learning and deep and their scope of implement in NIDS, the study hopes to drive the discourse to a more secure future.

1.5 Scope of the Study

The reach of the network security research encompasses a series of dimensions ranging from the intricacy of data encryption to the structure of secure networks. Despite the implicit learning being mainly concentrated on Network Intrusion Detection Systems, the scope of the topic includes a range of the limitations that are required to ensure a strict focus and a comprehensive examination.

With this concept in mind, this research aims to discuss and compare the four chosen machine learning and deep learning algorithms. They include Random Forest and XG Boost for machine learning, and LSTM and GRU for deep learning. The reason that prompted choosing the above-discussed algorithms is primarily the high probability and efficiency ratio in the field of intrusion detection. Furthermore, they each have distinct features and components to start learning and introducing the range of network traffic analysis and anomaly detections.

Likewise, it is also important to notice that the chosen dataset is an equally significant component that influences the result of any research. One of the reasons we use the HIKARI-2021 dataset is because of several factors:

First, what distinguishes HIKARI-2021 from most of the other datasets is that the majority of datasets are based on artificial data or using old traffic data; in contrast, HIKARI-2021 has a genuine lab-created data. Considering that it is created using genuine data, all the behaviors, patterns, and the anomalies in the dataset are close to the actual inquiries raised by the practical world system.

Second, it is relatively very completely fulfilled dataset, that is it has a myriad of benign and detrimental behaviors in the network. This will provide an appropriate

ground to line and examine the chosen set of rules in detail to find out what capacity they have.

Lastly, the creation was a result from 2021; thus, it will be one of the most relevant datasets for modern network behavior and threats; hence, it would provide results that would be relevant toward addressing the network challenges of the day.

1.6 Significance of the Study

The digital age we find ourselves present in cannot afford to have unsecure networks. Every piece of data, from a simple text to a critical financial transfer, requires the integrity of a network. It is within this context that this study shall be invaluable as it conclusively establishes the urgency of improving the efficiency of Network Intrusion Detection System.

Firstly, the research into the practical applications of machine learning and deep learning algorithms present an entirely new viewpoint on the question of current NIDS capabilities. Researchers, network administrators, cybersecurity professionals, and other stakeholders can benefit from gaining an understanding of the primary features and constraints of algorithms such as Random Forest, XG Boost, LSTM, and GRU to improve the accuracy and decision-making processes that are relevant to form, implementation, and use.

Secondly, the fact that the HIKARI-2021 dataset was used ensures that the work is based on the present conditions. Cyber threats continue to grow proportionally to the development of new technology, and thus, new tools and methodologies to combat them must also emerge. Utilizing a dataset derived from real data from 2021 not only makes the conclusions theoretically intact but also actionable on a practical level.

Furthermore, the importance of the study is that it affects multiple spheres. For businesses, improved NIDS will mean more effective control over their confidential data, and many saved in terms of reaction, damage control, and

reputation. For end-users, it will inspire confidence in online activities. Finally, it sets an agenda for further work for those researchers who will focus on developing this topic.

1.7 Thesis Organization

The understanding of how this thesis is laid out is critical for swiftly transitioning through all the knowledge, perspectives, and findings shared herein. The thesis is structured coherently to be clear in distinct chapters, with each section dedicated to a particular piece of work. The following is how it is outlined briefly:

Introduction presents the topic of the study by providing a background on the subject, its context, objectives, and relevance. It gives an overview of the study by introducing the readers to the various topics it explores.

Chapter 2, the Literature Review substantially contributes to the study by reviewing the existing literature on Network Intrusion Detection Systems, machine learning, and deep learning. It identifies the gaps in the literature that the study addresses and places it appropriately in the academic context.

Chapter 3 details the research design and methodology that was used throughout the study. It focuses on data collection, analysis, and the processes followed during the study to obtain reliable and valid results.

Chapter 4 is the primary chapter of the study as it contains the findings from applying and testing the algorithms on HIKARI-2021. The findings are reported, including performance metrics and comparisons with other types of the study.

Chapter 5 presents the results of the study. The final chapter additionally discusses the implications of the findings on this study's review of Hadoop and the field's real-world applications. Lastly, it reiterates what was found in the study, as well as makes suggestions for future studies.

2 LITERATURE REVIEW

2.1 Historical Overview of Network Intrusion Detection

The idea of NIDS is not new, its roots date back to the early days of the computer networks. As different systems started communicating with one another and sharing data, it became essential to monitor these communication channels for safety concerns.

The first IDS concepts were developed in the 1980s and relied on rules set by the administrators. These dictated good behaviour patterns and helped the system identify anomalies that could signal a breach [1]. Although highly innovative at the time, they were limited in detecting the number and type of attacks based on the configurations made by the administrators.

The 1990s improved on the statistical-based NIDS models. These models used the existing data traffic to trace patterns and used outliers in such patterns as signals of an intrusion. As the networks grew in complexity, more vulnerabilities appeared. [2].

The 2000s, and to this day, a new era of ideas based on machine learning started emerging. These systems were able to learn new patterns of intrusions from the existing ones and identify novel attacks that went beyond the pre-assumed rules or statistical patterns. [3].

Nevertheless, the issue of providing accurate and comprehensive datasets remained a challenge in the NIDS field. As Fernandes and Lopes emphasized, the very productiveness of a machine learning model, and especially as an intrusion detection system, heavily depended on the quality and applicability of the dataset used [4].

2.2 Machine Learning in Intrusion Detection

The incorporation of machine learning opened the door to a whole new world for intrusion detection. Unrestrained by defined rules or existing patterns, the NIDS enabled a dynamic response to new threats, which was far more efficient as a defensive measure.

The early machine learning models in NIDS applied supervised learning methods, and the training required a labelled dataset to be feasible to use. For the models to work, the manufacturers needed to identify the benign and the malicious network behaviours. Thus, the trained models could then operate real time and recognize new occurrences of the classified patterns [5].

Nevertheless, the evolution of cyber threats rendered a passive defence using known attack patterns ineffective. Unsupervised learning was born. These algorithms, such as clustering, find patterns in the network traffic itself or detect anomalies, even if they have not been previously seen [6].

However, despite the noted prospects, machine learning for NIDS faces numerous challenges. One of the problems, emphasized by Fernandes and Lopes, is the requirement for high-quality datasets [6]. However, the conventional datasets are suboptimal for various reasons, including obsolete attack patterns, parallel traffic, and poor relation to real-life systems.

In this text, the HIKARI-2021 dataset in Fernandes and Lopes description is a standard dataset in this work. It is obtained from real data collected in a controlled laboratory environment and provides fair characteristics of contemporary network behaviour, both benign and malicious. Additionally, as already stated, this description also raised the issue of selecting features: “The feature size of the HIKARI-2021 dataset were substantially reduced with no downgrading of the success rate of the machine learning model. This helps reduce the time needed for processing and makes the algorithm sections fast [6].

Another significant thing insight from their study is the high precision rates achieved by some machine learning procedures when using a balanced sample of the HIKARI-2021 dataset. Indeed, as revealed by any rate above 80%, it is undoubtedly the efficiency of the used dataset; however, it is quite possible that machine learning algorithms may also be more efficient for intrusion detection than it is currently believed. [6].

All in all, it becomes apparent that Network Intrusion Detection has travelled across a long story from basic rule-based systems of the 1980s to the present day's highly developed machine-learning-helped models. Nonetheless, everything can and should be developed even further, as shown by the research of Fernandes and Lopes. [7].

2.2.1 Random Forest in NIDS

Random Forest, which is among the popular ensemble learning techniques, is a critical machine learning tool in various studies. Due to its use of more than one decision tree and bootstrapping technique, Random Forest improves prediction and eliminates the challenge of overfitting that is common in decision trees [8].

In the sector of Network Intrusion Detection Systems, the algorithm is dominant, having been used in various projects to manage big data and high-dimensional feature extraction. For example, in their project, Kaur et al. used Random Forest and provided a comprehensive understanding of the algorithm in NIDS. According to the study, the algorithm outperforms other machine learning in terms of detection techniques [8]. This happens because of the ability of Random Forest to assign weights to features and thus identify the most predictive data regarding baleful activities.

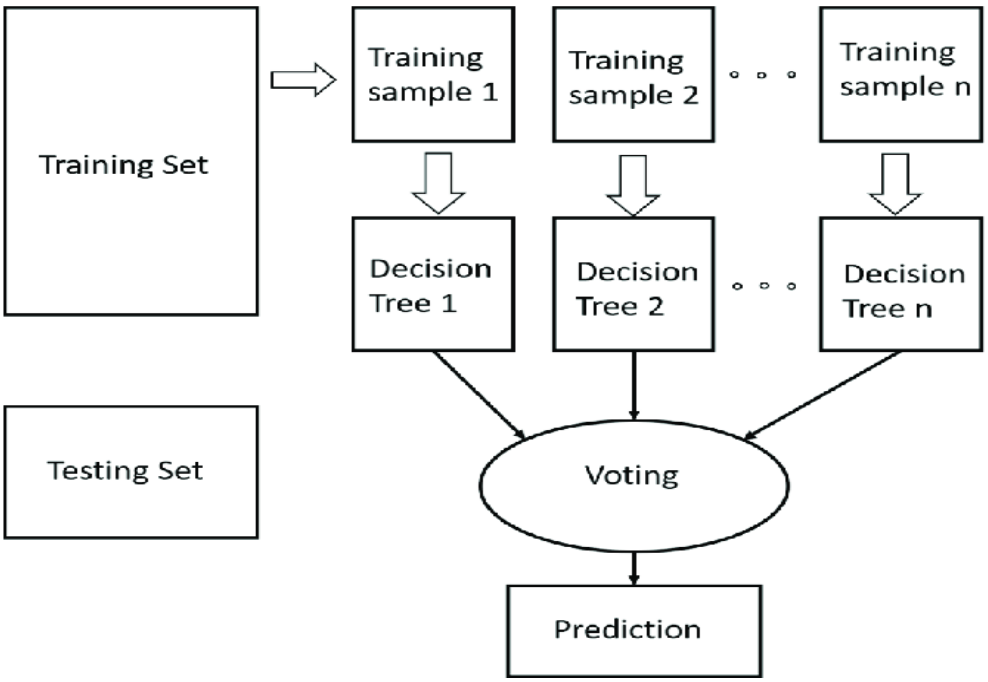

Figure 2. Random Forest Overall Structure [8]

2.2.2 XG Boost in NIDS

XG Boost stands for Extreme Gradient Boosting and is a high-level machine learning algorithm that has achieved popularity in numerous NIDS applications. It is an optimization of classic slope boosting that utilizes ordered practices to enhance performance and prevent overfitting [9].

In the realm of NIDS, it has been praised for its high adaptability and performance possibilities. According to a qualified experiment by Shiravi et al. [10], XG Boost, when used for the same purpose, it not only provides an acceptable accuracy but also requires significantly less training time.

Furthermore, Moustafa and Slay's [11] investigation has shown that even a properly constructed Random Forest algorithm can facilitate real-time detection with low false alarm ratios. Both aspects are critical due to the network sphere's nature; it is essential to have the ability to detect potential threats as soon as possible.

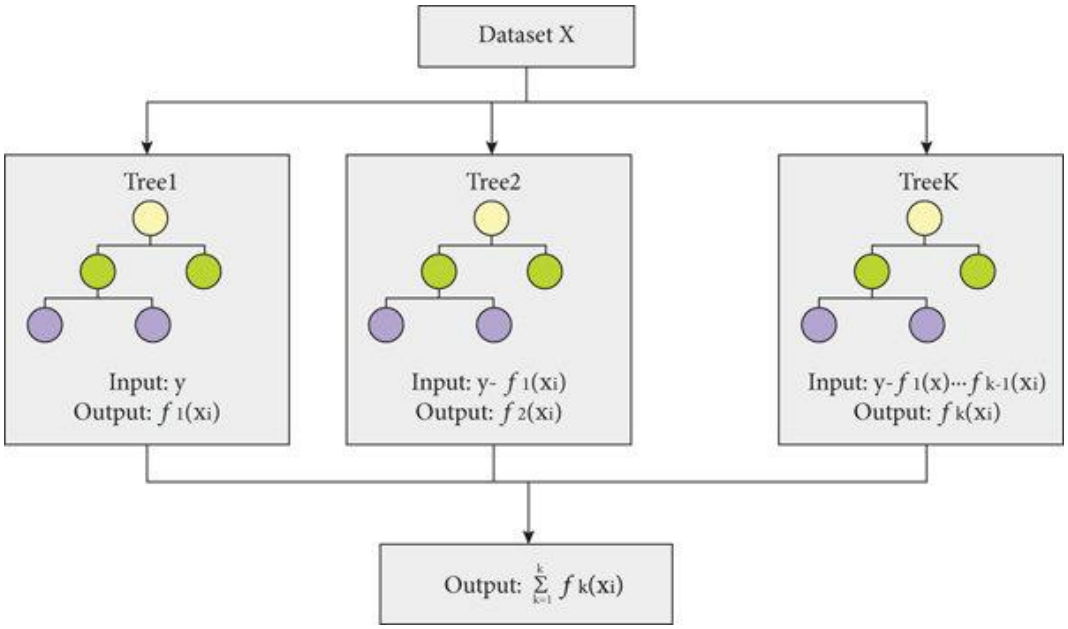

Figure 3. XG Boost Overall Framework [12]

Finally, Janarthanan and Macia [12] underscore the excellent performance inherited in the XG Boost algorithm when fed with uneven datasets, a constant phenomenon in NIDS research. According to the researchers, tuning the hyperparameters of the XG Boost allowed for the high rate of detection even when considered on the sparse side compared to good traffic cases [13].

Both Random Forest and XG Boost have become indispensable players in the vast land of machine learning-improved NIDS. Their superb efficiency in detecting intrusion signals confirms that the development of algorithms is an exercise that cannot be overstated regarding cybersecurity.

2.3 Deep Learning in Intrusion Detection

The discovery of deep learning, a type of machine learning that contains multi-level neural networks, has transformed a wide range of fields, ranging from image recognition to language technologies. Likewise, the great power of deep learning has boosted the typical Network Intrusion Detection Systems (NIDS) to offer more advanced cyber-threat identification [14].

Due to their hierarchical feature extraction procedures, which are based on surface and well-defined features, deep learning algorithms can expose deep and subtle patterns in network traffic, which is frequently outside the capability of traditional machine learning models. This is vital for recognizing sprouting and unidentified cyber-threats [15].

2.3.1 LSTM in NIDS

Long Short-Term Memory networks, which are a type of Recurrent Neural Networks (RNN), have proven highly successful in processing sequential data. Their architecture makes it feasible to maintain patterns over extended time periods, making LSTMs a possible candidate for time-dependent network traffic analysis [16].

LSTMs have already demonstrated potential in detecting dynamic attack patterns evolving over a lengthy time. For example, the key work by Borgnat et al. [16] has shown the LSTM's ability to capture long-term dependences in network traffic and enhance intrusion detection capabilities in situations of low-frequency and widespread attacks [17].

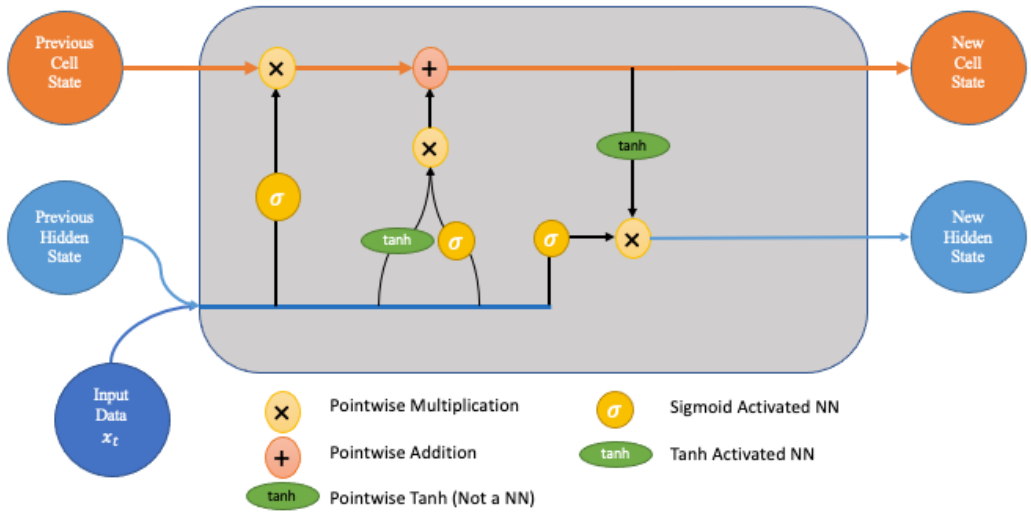

Figure 4. LSTM Overall Structure [17]

Moreover, Hafsa and Jemili [17] studied the versatility of LSTMs by combining them with other deep learning architectures. Combining them can establish hybrid associations that exceed LSTMs' detection rate. There is a strong recall that integrating LSTMs with models like CNN's can exhibit better or terrific results, particularly when dealing with many kinds of attack flows.

2.3.2 Gated Recurrent Units in NIDS

Gated Recurrent Units (GRU), another type of RNNs is one of the variant RNNs with similar but simpler architectures used for NIDS. Gated Recurrent Units GRUs possess a single update gate, which accommodates the forget and input gates in LSTM. Consequently, GRUs often deliver results of the same quality as LSTMs, though with less computational complexity [18].

Considering this potential, GRUs became a strong candidate for NIDSs, especially when functioning on huge sequence datasets. CADIA, experimental laboratory results discovered that GRUs, when optimized correctly, can deliver either similar outcomes or better outcomes as other NIDS DL models Pooled along while additionally significant efficiency [19].

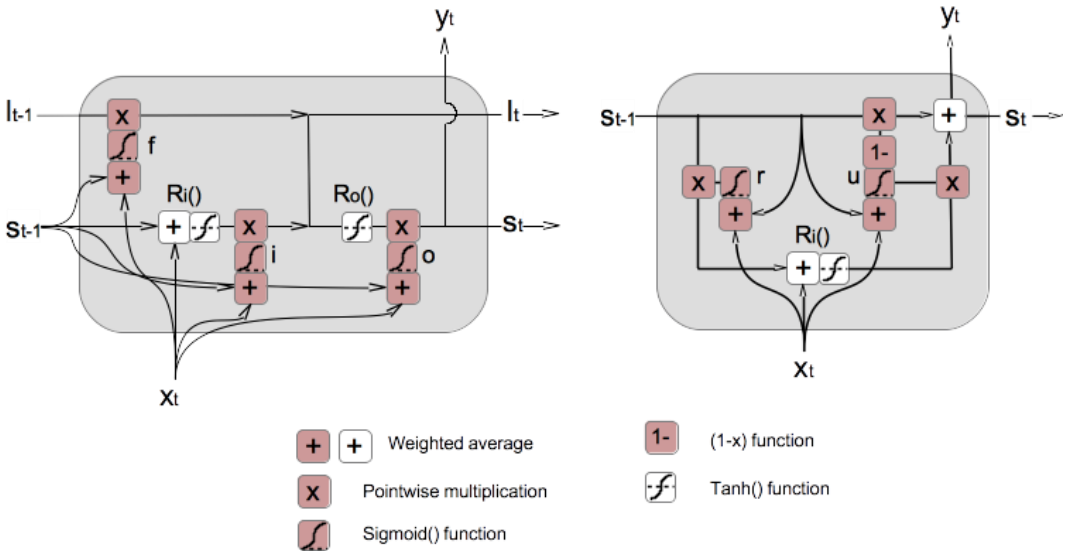

Figure 5. GRU Overall Framework [20]

Moreover, according to a study conducted by Jonker and King, GRUs are ideal for real-time intrusion detection, moving immense amounts of network data while maintaining accurate detection [20].

2.4 Comparative Analysis of ML and DL in Network Intrusion Detection Systems

The Comparative Study of Traditional Machine Learning Versus Modern Deep Learning algorithms in the implementation of NIDS algorithms has been the subject of many studies. ML with its fundamental algorithms has been the basis of intrusion detection solutions for many years. At the same time, old-fashioned neural network DL solutions are seen as strong competitors [21].

Commonly, comparison studies between ML and DL are conducted using various measures, such as accuracy, observation rate, incorrect-positive, and computational overhead that are often compared using standard ML algorithms and DL architectures. For instance, Anderson et al. compared Random Forest – an ML algorithm – with LSTM – a well-known DL architecture. The authors found that even though Random Forest processed faster and was more interpretable, LSTM was able to detect more complex and novel problems; still, it required more computational power [21].

Another well-known study completed by Sabahi [22] compares ML and DL indicating that DL algorithms are beneficial because they can automatically extract a raw feature representation from complex data, resulting in a more generalized and robust model. Still, the authors further indicate the application of DL and ML is based on different conditions that the NIDS require, including data size, the diversity of data attacks, and fundamental mathematical limitations.

2.5 The HIKARI-2021 Dataset for Intrusion Detection Studies

In the quest to improve the performance of intrusion detection models, the nature and quality of datasets utilized are crucial. Although HIKARI-2021, developed by

Ferriyan and his colleagues, is a significant leap, it is crucial for its focus on the importance of traffic encryption [23].

Prior datasets were based on non-encrypted traffic, creating a gap in research that focused on traffic encryption analysis. HIKARI-2021, with its labelled encrypted traffic, sought to close this gap and enhance NIDS development to accommodate the increasing use of encrypted communication.

What makes the HIKARI-2021 exciting is the combination of both real attack traffic and synthetic attack traffic. The synthetic traffic was meant to simulate real-world cyber-attack situations, and its addition to the genuine traffic significantly boosts the dataset's diversity and representativeness [23].

From the possibilities offered by the dataset that one would be able to make a full payload capture, one can only hope that it becomes a fact soon. As traffic encryption continues to become more common, being able to capture and analyse payloads without decryption, which is a high-latency process, would be critical to enabling real- time intrusion detection.

The Fernandes and Lopes study reviewed above is one of the first papers that took advantage of HIKARI-2021, focusing on classification tasks and underlining the potential future use of this dataset for enhancing NIDSs, both ML and DL-based [25].

All in all, datasets such as HIKARI-2021 are crucial for the progress of IDS research, ensuring that the models are constantly updated to capture insights on how to maintain relevance and robustness in a changing environment.

2.6 Obstacles and Constraints of Existing Methods

Despite the vast evolution of NIDSs domain with a Machine Learning and Deep Learning exploitation, a particular collection of issues and constraints frequently reappears. Understanding these difficulties on the background of the practical

experience can be beneficial not only for fine-tuning the current techniques but also for indicating the future of research and development.

First, one of the most severe issues, particularly for ML-based methods, is overfitting. It occurs when a model learns the training set well but fails to generalize on a test or unseen one [25]. It is also a problem for Deep Learning methods because of the vast number of parameters, thereby regularizations are not optional.

Given the rapidly growing amount and complexity of network data flow, NIDS solutions, which can scale up, are indispensable. While DL models, including deep neural networks, show potential, their processing demands generally preclude their use, especially in real-time scenarios [25].

ML models, including Decision Trees and Random Forest, offer some level of interpretability – a network engineer examining the nodes of the RT can understand why a particular rule was proposed. In their turn, complex DL models, such as LSTMs and GRUs, are called “black boxes” because the way they arrive at their decisions is practically impossible to reconstruct [24].

For one, cyber adversaries consistently innovate, creating new attack methodologies. Such development frequently gives rise to zero-day vulnerabilities NIDS are doubtful to diagnose, particularly when relying on old training data [26].

Additionally, networks datasets rarely involve balanced training instances, as benign instances far outnumber outlier ones. Consequently, trained models perform favourably in terms of accuracy metrics but are not efficient at genuinely identifying attacks [26].

Commonly, the usage of true network traffic for training raises privacy concerns. Although many network datasets are anonymized, the usage of encrypted traffic for training is challenging: the interpretation of the message payload is not simple

itself, but, more crucially, the use of it for training might also be perceived as intrusive, breaching measures like GDPR [26].

2.7 Cloud Based Technique Challenges and Advantages

Cloud Intrusion Detection Systems are a critical tool for safeguarding cloud environments and data and infrastructure from the sophisticated cyber threats. The deployment of IDS in cloud-based environments contributes to mitigating the threats and challenges faced by cloud services like hacking, data leakage, unauthorized access, and insider attacks. According to Bharati and Tamane [26], IDS is continually changing to match the new cloud-based environments and increase the flexibility of the systems to counter future security problems. This finding is critical because it offers an understanding of the future of cyber threats in cloud computing and the need for revolutions in detection techniques.

The need to ensure the integrity and availability of cloud services requires developing reliable mechanisms for fault tolerance in cloud-based IDS. Although presented in a slightly different context, the high relevance to the maintenance of continuous work despite system failures and attacks is demonstrated by the research of El-Desoky Ali, An. Hisham and Abdulrahman A. Azab [27] on a framework for shared computing on desktop grids. Moreover, the importance of adaptive response strategies is proven in the study by Azab and Kholidy [28] on adaptive decentralized scheduling mechanism.

Despite the limitations of their knowledge of cloud technologies, information security specialists generally agree on the basic principles based on the NIST Special Publication by Bace and Mell [29]. These principles include the fact that IDS" monitors, analyses, and responds to unauthorized activity". Therefore, they can be used as a basis for designing a cloud-based IDS that can handle the complexity and large scale of cloud infrastructures.

Another in-depth formal classification proposed by Debar, Dacier and Wespi [30] also includes intrusions detection frameworks. This proposed classification enables the grouping of attacks and the methodologies to detect them in a unified approach. It helps to develop the targeted detection algorithms that are capable to reveal and neutralize certain types of threats exclusively in cloud environments.

Moreover, the model constant interruption recognition master framework Lunt and Jagannathan (1988) and ASAX framework Habra et al. (1992) can involve in the rule-based and anomaly methods. These frameworks could train engines to distinguish between regular behaviour and abnormal activity, that is critical for the current IDSs.

2.8 Overview and Areas for Further Research

A review of the composition on ML and DL in the context of NIDS outlines several milestones, innovations, and subsequent iterative improvements. However, it also highlights several gaps that provide avenues for further research.

While datasets such as HIKARI-2021 are a quantum leap, a constant need for recent datasets is essential. They should include topics of modern network traffic, including all its peculiarities, tendencies, and emerging assault vectors [26]. Even while the precision and detection rates are top priorities, ignoring the trend toward increasing model visibility is far from an option. Approaches that can raise the curtain on advanced DL models “black box” nature are beneficial in actual deployments [26]. The literature frequently portrays ML as opposed to DL, highlighting their positive and negative aspects. However, there is a noticeable lack of information on the available hybrid models that can stabilize the positive aspects of both design principles [23].

Especially, as the digital space moves towards increased encryption, it becomes imperative to understand and implement NIDS models for encrypted traffic. For instance, there appears to be a lack of full-spectrum studies that can address the

challenges of encrypted payloads holistically [23]. Most current models that might prove to be effective in controlled conditions will perform suboptimal in real-time due to imprudent computational limits or lack of scalability. Therefore, the subsequent research that focused on dynamic adaptation of these models to meet real-time requirements would be beneficial [23]. Ultimately, NIDS using ML and DL is a rich field of study that requires further examination, either for fine-tuning existing models or developing new ones.

3 RESEARCH APPROACH

3.1 Research Methodology Overview

The Research Approach acts as a foundation for the research, helping to maintain structure, rigour, and reliability. For instance, this study, comprised a combination of qualitative and quantitative methods to apprehend the topic fully.

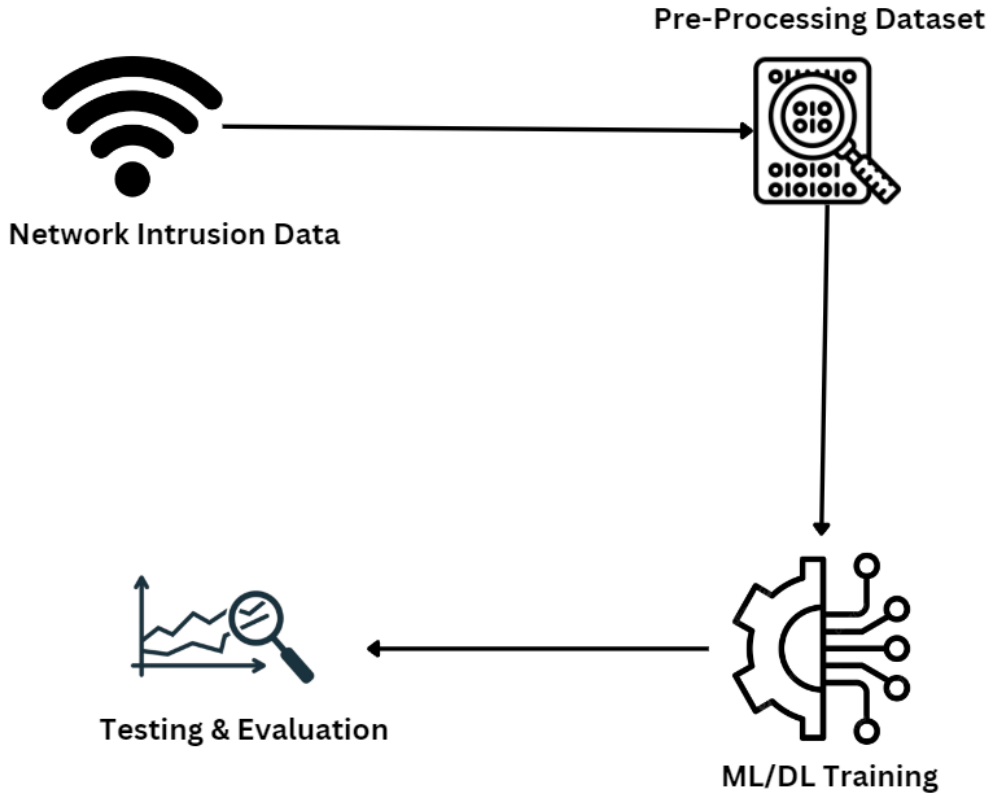

Figure 6. Formulation of Problem Statement

At the core of the research was HIKARI-2021 as the dataset. As previously discussed, this collection is based on actual network behaviour, so it will provide enough data to translate.

Prior to conducting any evaluation and assessment of algorithms, the collection was pre-processed. This made it certain that everything but noise, unimportant

details, and unusual patterns were removed as they could alter the end product. Normalization, feature extraction, and data scaling would all be conducted.

Following the preprocessing and data preparation of the dataset, the four chosen set of algorithms, namely Random Forest, XG Boost, LSTM, and GRU, were subsequently tested. The testing was carried out through the four algorithms interacting with the data, dividing it into sets for training and evaluation. The algorithms learned the patterns based on the sets of commands and made predictions on the testing data, based on which their speed, accuracy, and predictability could be evaluated.

Once the testing of the algorithms was complete, all algorithms and their performance were compared using a table form comparison of the performance metrics. A comparative analysis of metrics was the next step of the methodological framework's implementation.

Based on the discoveries identified and provided by the comparative analysis, the blended design was created. The model tried to merge the most advantageous pieces of the machine and deep learning procedures discussed above, and it is intended to provide a detailed solution for invasion detection.

The generated blended model goes through multiple tests and iterations. All performance evaluation metrics were meticulously assessed, and several improvements were proposed to boost its effectiveness.

In the end, the research arrived at the final results. The accomplished conclusions were supplemented by a set of recommendations concerning potential use cases, further research opportunities, and areas of deploy ability.

3.2 Data Collection

The basis of most modern Intrusion Detection Systems (IDS) is comprised of data-driven techniques. Therefore, the importance of ensuring the necessary credibility

and quality of data cannot be overstated. The HIKARI 2021 dataset as well as data pre-processing and cleaning are introduced next.

HIKARI-2021 is a state-of-the-art assembly that was created to remedy the most glaring drawbacks of existing datasets – most of them do not contain the encrypted traffic data that has become increasingly prevalent in contemporary networks. With a combination of real and synthetic network traffic, this dataset incorporates a variety of features that enable capturing numerous nuances of the network flow, thus creating a complete image of the existing traffic patterns. Given the transition to encrypted traffic taking place in many environments, the significance of the dataset designed to provide support for this paradigm shift can hardly be overstated.

Before feeding complex models, the dataset must be cleaned up to remove any anomalies or redundancies which could otherwise distort the findings to follow.

Removal or replacement of instances with missing feature values to keep the dataset intact. The values of features should be scaled where required, particularly those that have a high dispersion or scale; otherwise, the model may overemphasize certain features. Data Transformation: For instance, the presentation code converts categorical data to a format by one-hot or label encoding. Statistics could be used to identify and remove any anomalies, that is, those instances which may potentially become a source of the distortion while training the model.

3.3 Attribute Development and Selection

A science as well as an art form, feature engineering is the process of creating new attribute from current features with the help of both professional expertise and learnings from preliminary examination of data. Meanwhile, attribute selection involves selecting solely the most crucial features in the dataset, removing the rest less useful and the harsh ones. The data was normalized to place all factors on the same scale. To guarantee that variables with larger variances had just as much

impact on the study as the rest, data scaling was utilized. Additionally, obtaining certain attributes algorithms were also employed to extract the most significant attributes from the dataset.

To refine our dataset and maximize the execution capability of our intrusion detection structure in terms of prediction, we carefully implemented several techniques into the Feature Engineering and Selection component, which is the concern of this chapter. Following is the heat relationship matrix of the Dataset:

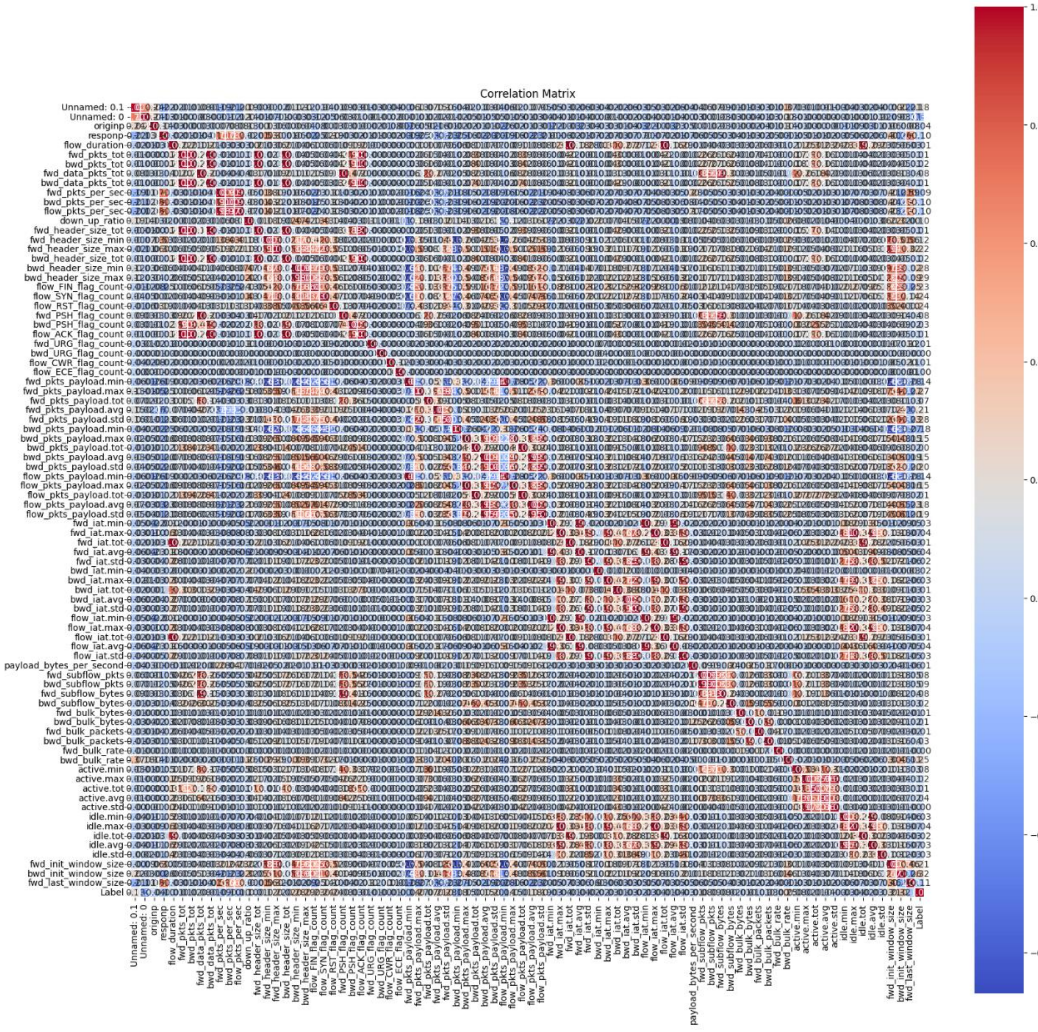

Figure 7. Heat relationship of the Dataset (HIKARI-2021)

Initial cleaning and preprocessing of the dataset. Given that the dataset constitutes network traffic data, it was highly important to fill in missing values,

outliers, and highly irrelevant features that could significantly skew the prediction accuracy of the model. This is fundamental in enabling the dataset to be as clean and credible during the prediction phase and model training.

Different features have distinct values throughout the dataset. Therefore, normalization and scaling were employed to equalize all the variables to a common scale. This factor is significant in ensuring that one variable does not overwhelmingly influence the decision of the model since it has a greater magnitude. The Min-Max scaling approach was put to work and the adjustments made in terms of the range are 0 to 1, which is relevant for the algorithm that is more sensitive to the magnitudes of the data used.

The selection of features was accomplished through the selection of the most meaningful ones effecting the detection of network intrusions. A range of filters, wrappers, and embedded approaches were used to assess feature importance. Filtration methods were chosen due to their speed and simplicity since they were based on the evaluation of features by their correlation to the target variable, implemented by certain statistical measures. Wrappers were implemented for optimization purposes in order to find the most precise combination of features by evaluating several possible combinations based on accuracy measures. Embedded solutions were those that were integrated into algorithms and allowed for feature selection during the modelling process with special regards to the performance of the algorithms.

Since network traffic data has a very high dimension, reducing the dimensionality was essential as it simplifies the model without losing important information. We used Principal Component Analysis (PCA) to minimize the dimension so that the original features can be replaced by a reduced set of independent variables termed as principal components. This increased computation but also may improve the model by focusing on the important variation in the data.

Using domain knowledge and analysis of exploratory data analysis (EDA), we created features that summarize the data and capture much information about the network traffic patterns. This includes interaction features, aggregations, and polynomial features as they may identify patterns in the data that indicate intrusion. Our transformed dataset provided our model with more useful information from the input because of feature manipulation.

Since a dataset has multiple categorical attributes, converting these columns into numerical representations is crucial for the model to learn from it. For the ordinal variable, since the data is arranged in a natural order, we used a Label Encoder, and for the nominal, data we used One-Hot Encoding to create a binary matrix of the categorical data.

3.4 Model Development

Model Development is a central part of data-driven analysis. The choice of algorithms should be confirmed by a good level of implementation that will match the peculiarities of the dataset. In the context of our research, four models have been developed: Random Forest, XG Boost, LSTM, and GRU. Each is unique to an element of the intruder detection system and has its advantages.

A version of ensemble methods, the Random Forest, is based on the construction of a decision tree “forest.” The general solution is represented by the many trees. Every tree in the forest was generated from a bootstrapped sample, which was a sample taken from a training set and selected repeatedly. In the implementation of this model, there were 100 trees. “Gini” was the division criterion which was used to determine the quality of a division in the dataset. Every tree considered a random subset of features while multi-splits solving, which made the model diverse, avoiding overfitting. The simplicity of implementation, the option to do such with a large dataset with a higher dimensionality, the option of missing values treating achieve a solid result to the corrected version of hyperparameters allows for the highest quality indicators.

One of the optimized gradients boosting libraries is XG Boost, which is short for extreme Gradient Boosting. XG Boost is faster than other boosting algorithms. It can raise trees simultaneously. With our study setup, when implementing the XG Boost model, I performed no modifications to default hyperparameters. However, during the model training and testing, I aimed to maximize the model's learning and tree depth by applying a hyperparameter tuning algorithm. Thus, for instance, the model's learning could be adjusted to mitigate overfitting. The tree depth allows one to maintain a simple model. To avoid overfitting, the regularization parameter was implemented. One of the most valuable features of the XG Boost model is that it may handle sparse data. This is an issue since the network data has many zeros.

An RNN architecture, Long Short-Term Memory, is one of the kinds of LSTMs. LSTMs are ideal for processing the data sequence upon which the network traffic structure is based since they have feedback connections rather than feedforward connections like typical neural networks. Before employing an LSTM model, the input data had to be reshaped to mirror that of an LSTM network. In summary, the model we constructed had an input layer, a dense output layer with a SoftMax activation function, and an LSTM layer with 50 units. We used an "Adam" optimizer and "categorical_crossentropy" loss because we have numerous classes in our dataset. The major advantage of our LSTM network is that it can learn from long-term sequences, which is difficult for an ordinary neural network to accomplish. It is highly appropriate for our time series data as a result.

A Gate Recurrent Unit is a different form of RNN, even it is working comparable to LSTM. GRU lowers the cost of the network computation by using only two gates: a reset gate and an update gate. Our utilized model in which accurate architecture used for LSTM. We had reshaped the input data, and before creating a model, we assembled an input layer with a 50-unit GRU layer. The output layer consisted of a dense layer with SoftMax activation. We compiled the model in the same way as LSTM. The main advantage of GRU is its simplicity. It is easy to understand and

simpler than the LSTM, which makes it a better option when training time and model complicity matters.

3.5 Verification and Evaluation Approach

An excellent testing and validation plan are the main component of an effective framework. In our case, we considered the hold-out validation approach. It means that the entire dataset was chosen randomly and then divided into two smaller datasets. One part was the training set, where the second part was 80% of the dataset, and the testing set – the remaining 20% was used to calculate the performance. Until the model was tested, it did not see numbers from the training set. It was a good way to check whether the model operates in an appropriate manner with new, hidden data. Accuracy was one of the performance measures we used to determine the efficiency of such models. Confusion matrices were also used as a tool in the same regard and the same was plotted to visualize this state for each model.

Consequently, the models were developed and validated conscientiously and using the best standards. The different models ensured that the intrusion detection problem was tackled from various angles.

3.5.1 Dividing the Data: Training, Validation, and Test Sets

Dividing the data is the first step in any machine learning or deep learning project. The task of splitting data is to ensure that the model learns from one subset of data and then tests on the second set to assess how good the model generalizes.

The dataset used in our study was partitioned into the following set of scales:

Training Set (70%): The pattern recognition models are heavily dependent on this section of the data. With the help of training this part of the model, the models learn the specifics, links, and patterns of the situation.

Validation Set (10%): The role of this set is to tune and refine the models. As main training occurs, the models are tested with validation set so that in this way we are ensuring that models do not leer the training data but are generalising good the information to make predictions.

Test Set (20%): Once the model is trained and adjusted, the evaluation of the model occurs. The following section of the data set has nothing to do with the system prior to the evaluation of the system.

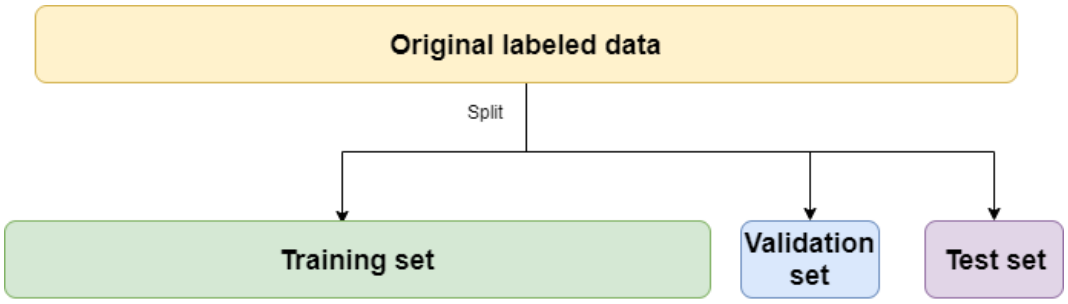

Figure 8. Training, Testing and Validation Splits

This computation ensures that the models can generalize to predict entirely new data rather than just the established patterns.

3.5.2 Performance Measurement Criteria

The primary assessment criteria metrics selected to measure the performance of our models are:

Accuracy: Accuracy is an indicator that emphasizes the percentage of predictions the system actually predicted. It is determined as a percentage of accurately forecasted patterns to total data examples and offers a high-level summary of model ability.

Confusion Matrix: A misinterpretation matrix is a table employed for a description of the model using classification performance. It gives the true positive, true negative, false positive, and false negative in the matrix equation. Our confusion matrix analysis shows me what specific types of mistakes our model makes.

3.6 Comparative Analysis Approach

An Intrusion Detection System is designed to create a secure cyber environment by detecting malicious behaviour. A comparative analysis is a crucial strategic approach in determining the best IDS's approach to be used. This section discloses our strategic approach when it comes to carrying out a comparative analysis of the machine learning and deep learning analysis models such as Random Forest, XG Boost, LSTM, and GRU respectively.

When evaluating ML and DL models, one must have a set of criteria. In this case, assessment criteria like precision and confusion matrix were used to establish the actual performance. Therefore, the model with a high accuracy rate and good confusion matrix results has higher that for the raw performance.

While DL models can identify complex patterns, this can come at a cost of high complexity. Alternatively, the simplicity of the ML model may be appealing. The simplicity of the model in terms of time taken to train can as well be tough. In that view, while it might be easier to train a DL model, Compared to ML models, one takes a long time to train. Equally as important is the performance of a model constructed with new, untested data. The reality is that the model may not operate well on new data, even if it has a high prediction accuracy on the training set. A model should also not be a "black box". For example, while it is difficult to understand how a neural network comes up with a decision, it is easier to understand models such as Random Forest or XG Boost. The presentation of a model that can be a "black box" is not useful in model interpretation. While an ML has it sometimes needs to be understood, an ML model goes ahead of the model that can be a "black box".

In conclusion, evaluation, data splitting, and comparison of models are systematic. It is important to follow the drawn procedure when meeting the best model fit for space usage. Predecessor analysis of ML and DL models reflects the ability of the two approaches to secure detection.

3.7 Limitations of the Methodology

While our research methodology was carefully constructed, it naturally has some limitations that must be acknowledged in the spirit of maintaining the integrity and transparency of the study. The first and one of the most significant limitations concerns the datasets used. While we employed the datasets widely used and labelled in this field, they might not fully represent all possible real-life situations. In some cases, the real-life data differ quite substantially from standardized datasets and introduce unforeseen intricacies and challenges that our models, trained on the standardized set of data, might not be particularly well fit for. Moreover, even at the time of their superiority compilation, datasets do not describe the most recent threats and vulnerabilities. It can be expected that new threat vectors may arise, and the landscape of cybersecurity threats may change over time, meaning our algorithms would not suit for an accurate identification of them. Finally, a common problem in many cybersecurity datasets, it class imbalance. Many sets are imbalanced and are dominated by the positive class. This can bias the models towards the major class, substantially lowering the prediction accuracy for the minority class, representing actual threats.

Another limitation concerns the models themselves. Complex models, especially in deep learning, sometimes tend to overfit the training data. This results in models that are overfit and do not perform well when introduced to new, previously unseen data. Additionally, many deep learning models suffer from interpretability. These models are highly complex and behave as black boxes were discerning the logic behind any particular prediction of decision is impossible. Interpretability issues are important in many fields where decision-making logic must be transparent. Finally, some models might be too computationally expensive to use in real-time or resource-limited scenarios.

Although strong, the described evaluation methodology has several limitations. The primary limitation, however, remains the potential ignorance of some important performance indicators. In other words, the focus on selected metrics,

such as accuracy and the confusion matrix, may forget about other important metrics like precision, recall, or the F1-score. This is particularly important for the scenario of class imbalance, as the performance may differ significantly across other datasets. Algorithms not being generalized, it would probably have different results on a fresh, not been seen dataset, aside from the evaluation, which has been described. Considering the computational limitations presented in this study, it is also crucial to mention the limits and the influence observed in this study. More specifically, due to time and resource constraints, not all hyperparameters and model architectures have been investigated, and the best solution may have not been selected.

An important consideration of this study and, in general, involving computational algorithms and systems, is the temporal nature of methodologies. It should be noted that cyber threats and their characteristics evolve very quickly. What works now in just a few hours can become irrelevant and, therefore, ineffective. Therefore, it is important to revise the methodology, especially in the field of cybersecurity, and review it.

3.8 Summary

This chapter has provided a detailed investigation of the research methodology from initial selecting and execution of machine learning and deep learning models to their application in detection of threats. Still, an overall interpretation of these findings must be read and understood considering the expressed limitations. The ever-changing challenges of the cybersecurity field require a constant leveling up in working and model refinement. This study, as extensive as it is, offers a snapshot in time on this evolving timeline. It offers a trustworthy basis while highlighting the potential and achievable goals in machine and deep learning roles in intrusion detection. Current cybersecurity advancements necessitate a change in how we tackle problems- a need for further exploration inside and outside the errors. The current achievement and the need to improve both have been demonstrated, a

call to further innovate while understanding the need to adapt model use in more challenging times.

4 RESEARCH RESULTS AND ASSESSMENT

4.1 Overview of Experimental Configuration

The desire to obtain high-quality insights into intrusion detection through the two learning algorithms required an extensive experimental setup. Such a design offered credibility through rigorous research forms. The major components of our experimental design that facilitated the entire process were the computer system, which comprised of hardware and software, carefully selected, and prepared as well as the data configurations that provided a holistic opportunity for the two models to learn, validate, and enhance the model's skills in identifying intrusions.

Our research configuration relied on a computing environment with certain technical Specs and software specifications. The primary hardware and software utilized in this experiment included a computer with a multi-core processor. The multi-core processor was vital as it enabled our experiment to make use of parallel processing, which is necessary when training deep learning models, which are computationally intensive. Secondly, the computer had 32 GB of RAM, which was essential in ensuring the smooth processing of data, particularly with data sets hundreds of gigabytes in size. The third hardware device used was an NVIDIA graphics processing unit, which positively impacted the training times of our deep learning models.

On the software side, we used the Python programming language, which is a versatile programming tool with vast libraries supporting several machine learning and data analysis activities. The libraries used are Scikit-learn for machine learning, TensorFlow and Keras for deep learning, and Pandas for data manipulations and preprocessing and Matplotlib and Seaborn the latter being vital in making additional comprehensive distributional plot assessment.

4.2 Analysis of Dataset Distribution

Data is the essential component for every machine learning or deep learning model. The HIKARI-2021 dataset was used in this study. The HIKARI-2021 dataset presents the challenges and complexities that in the field concerning data cybersecurity. The HIKARI-2021 dataset avails different types of cyberattacks and patterns of network connections used to carry out the attacks.

4.2.1 The HIKARI-2021 Dataset's Descriptive Statistics

First and foremost, we sought out the descriptive statistics for the HIKARI-2021 dataset. The multiple columns in the dataset allow us to deep dive into various characteristics of the network traffic. The columns that contain the origin IP address with port are braked with two features: the former utilizes the columns originh and originp, and the latter employs the columns responh and responp to pinpoint the response IP address with port.

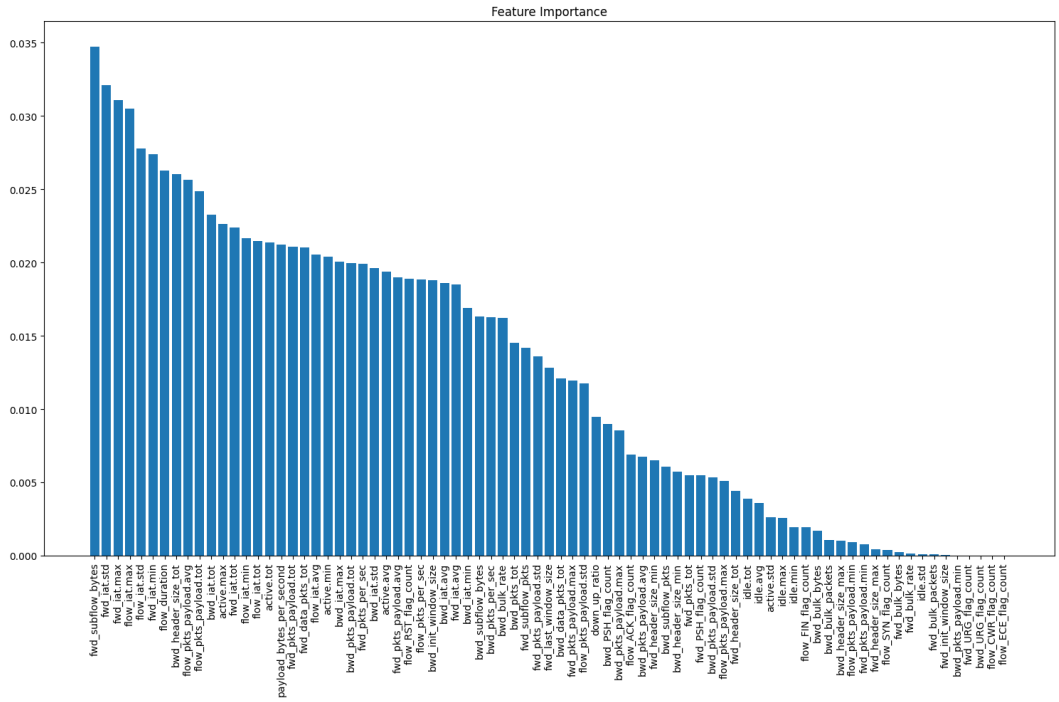

Figure 9. Attribute Significance Analysis of Hikari Dataset

The initial observations of the dataset relate to network flows with some critical properties. In particular, flow_duration, fwd_pkts_tot, and bwd_pkts_tot mean the flow length, the total quantity of forward packets, and the total quantity of backward packets, respectively. The dataset consists of a variety of different flows, some of which have a high value of packet rate, as shown by the fwd_pkts_per_sec and bwd_pkts_per_sec values obtained during the initial data exploration.

Moreover, a variety of flow-related flags have contributed to the following: flow_FIN_flag_count, flow_SYN_flag_count, flow_RST_flag_count, which are the flags for resets, synchronization, and termination in this order.

However, traffic_category and Label were the most crucial factors in the dataset. The former describes the type of traffic while the latter, which we use in our study, describes the safety of the flow as a category, as a threat, or as neutral to the field of security, e.g., “Bruteforce-XML”.

4.2.2 Classification of Attack Types Distribution

A detailed exploration of the analysis methods followed the preliminary exploratory data analysis. The focus was on understanding the range and allocation of attack classes in the collection. It is evident that the greater the proportion of attack classes in the dataset, the more robust the training, validation, and testing results.

Based on our preliminary analysis of the HIKARI-2021 dataset, the attack classes were of various types, including “Bruteforce-XML”. A visualization to examine the frequency distribution of each class was subsequently employed to provide an overview that illustrated the frequency distribution of each type. Through visualization, this exploration provided an understanding of the most common attacks and the existence of class imbalance.

It is interesting to note that some attack classes were well-represented, while others were less represented. Such disproportionate allotment of instances made

the pre-processing procedures such as resampling or the synthetic data-generating methodologies necessary; otherwise, our models could develop bias during the training process.

Furthermore, we further explored the correlation between various attributes of the network flows with the attack class itself. For instance, we had insights into some flags or specific packet rates that might correlate more with specific types of attacks more. This knowledge was critical in ensuring that we personalized and customized the models to figure out the distinctiveness and characteristics associated with each individual cyberthreat.

Therefore, the HIKARI-2021 dataset was ready to be explored due to its columns, and even on the level of individual records. Our getting to know the data was predicated on the comprehensive discovery of the descriptive statistics and attack class distribution, so we already had an impression of what to expect from the modelling process. After that, the conditions were unlocked, the data was investigated, and the way to competent models of intrusion detection was prepared.

4.3 Evaluation of Separate Models

After the distribution of the dataset, the model implementation and evaluation became the following steps of the HIKARI implementation. The HIKARI-2021 dataset was implemented to train the algorithms Random Forest, XG Boost, LSTM, and GRU. The results of the algorithms' functioning were received by the application of the figures of confusion matrices. It was convenient to calculate the overall quality of the intrusion detection and the strengths and weaknesses of the algorithms.

4.3.1 Random Forest: Findings and Testing

The Random Forest model has an accuracy of 93.77%. This is supported by the fact that it was able to identify 99359 true positives; hence it is highly accurate when

identifying malicious network traffic. Similarly, the algorithm could predict that there were 4773 true negatives, and thus it is accurate in identifying a benign network operation. However, the model predicted 2717 false negatives and 4201 false positives. To minimize the number of counterfeits, it is necessary to focus on increasing the number of false positives, especially when applying a real cybersecurity application.

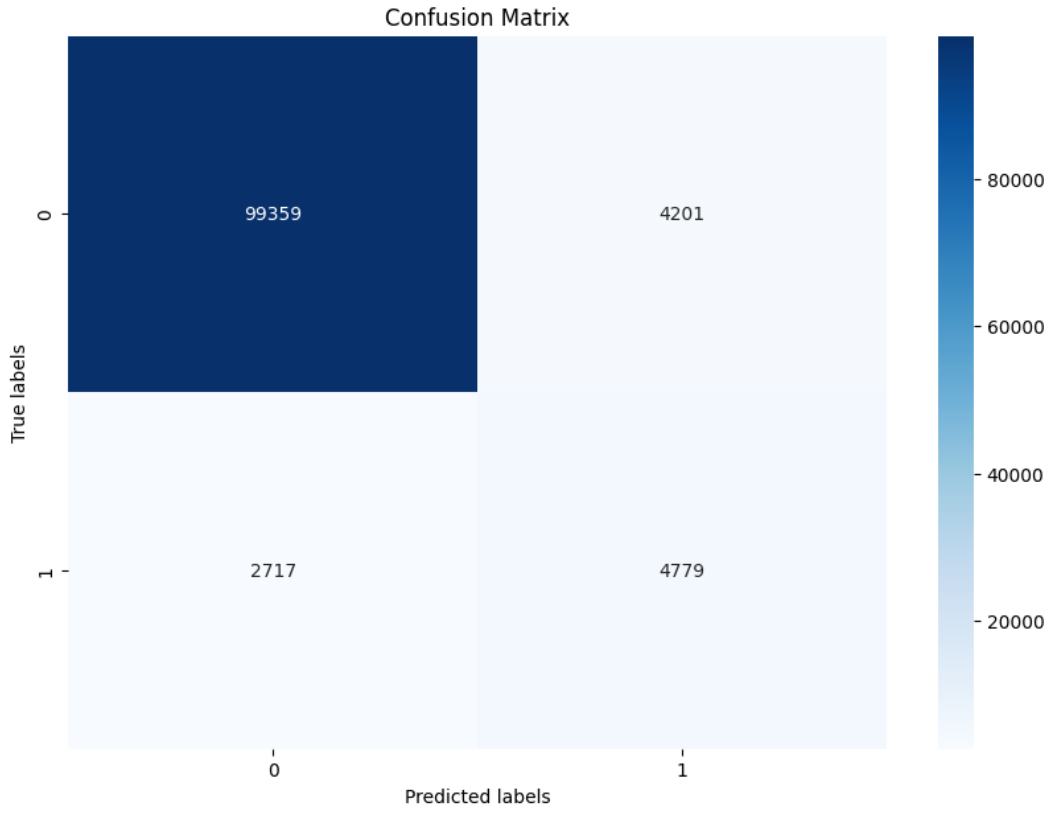

Figure 10. Matrix of Confusion for Random Forest

4.3.2 XG Boost: Findings and Testing

The accuracy of the XG Boost model was 93.02%. In the identification of fraudulent traffic, it showed slightly higher performance than Random Forest – 99888 true positives. In the case of benign front: 3418 true negatives. However, compared to Random Forest, the model produced 3672 false positives and slightly higher false negatives – 4078. Given the fact that the false negatives are too high and indicate possible security lapses, the model needs further calibration.

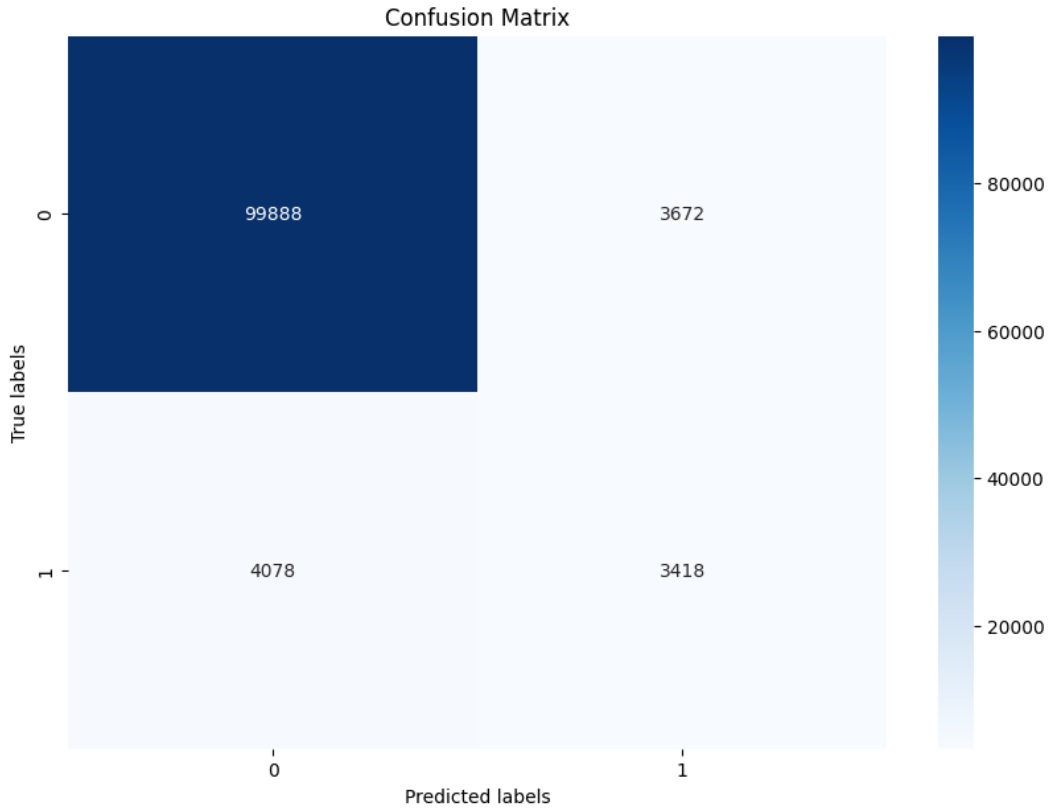

Figure 11. Matrix of Confusion for XG boost

4.3.3 LSTM: Findings and Testing

The deep learning version, the LSTM model, was 92.48% accurate. Interestingly, it has registered 88191 true positives, but its acquisition of 103280 true negatives was exceptional, demonstrating a strong ability to detect benign traffic. However, the 15360 false positives suggest that the model is likely to confuse benign traffic with malicious traffic. On the contrast, the false negatives were extremely low, with only 202 registered, indicating its effective performance in identifying genuine risks.

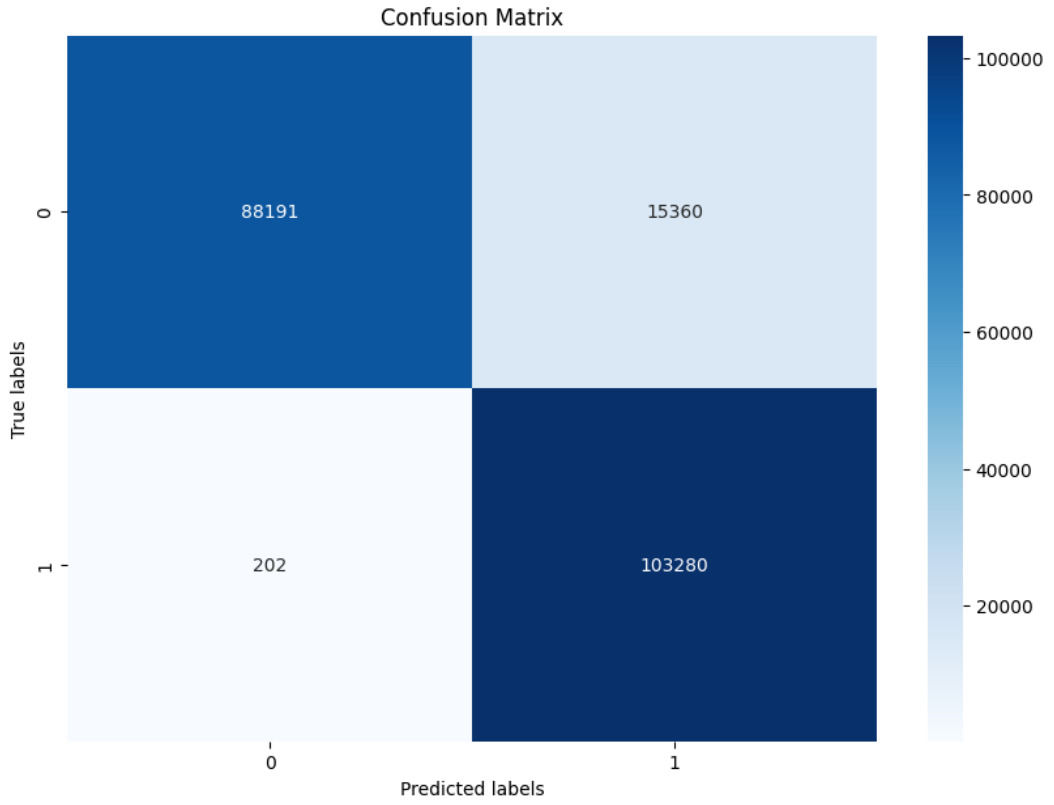

Figure 12. Matrix of Confusion for LSTM

4.3.4 GRU: Findings and Testing

The GRU deep learning model's accuracy, which was shallower than the LSTM, was 92.50%, nearly as high as the accuracy of the LSTM. It had 103203 true negatives and 88309 true positives on everything; however, it yielded 279 false negatives and 15242 false positives. It is not surprising that the GRU and LSTM behaved similarly in this respect, as both are deep learning models.

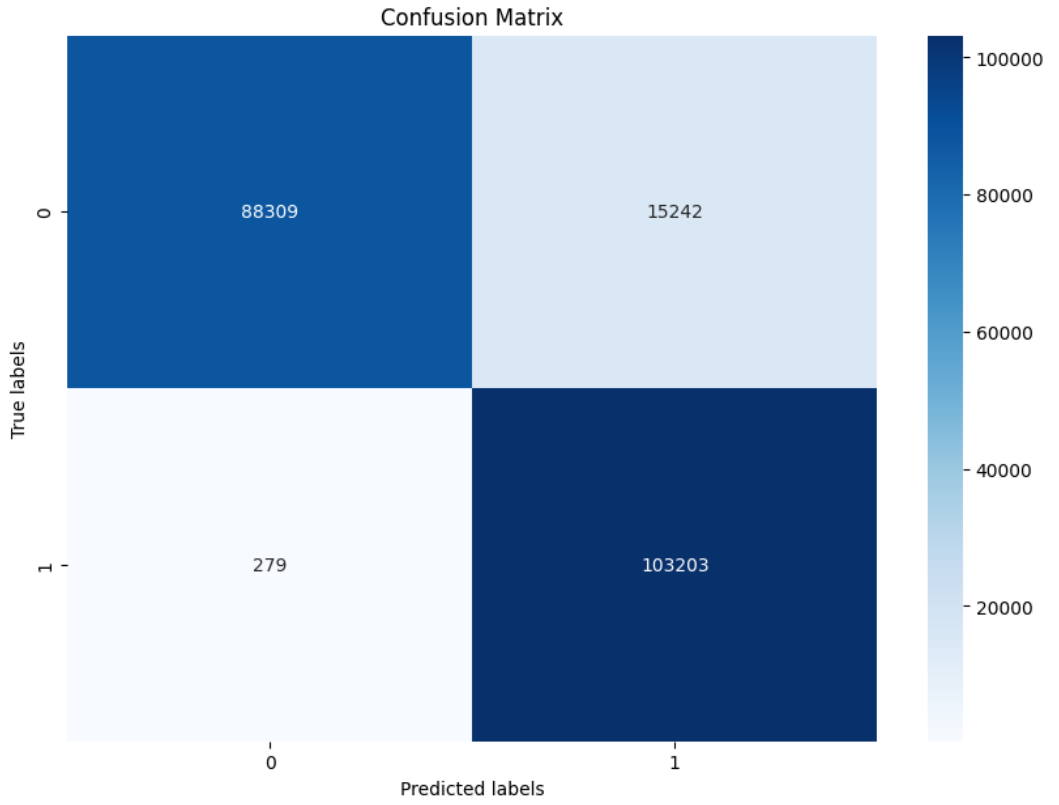

Figure 13. Matrix of Confusion for GRU

4.4 Comparative Assessment

Furthermore, a comparison was made to understand the performance of our model with the Benchmark. For all those specific models of KNN, MLP, SVM, and RF used the best with the intruding context presented by Ferriyan et al. [4]. Here, the accuracy was recorded very high like KNN with 0.98, while MLP and SVM were each 0.99; thus, RF was expected to have 0.99 matching.

Table 1. Comparative Assessment of Models

Model	**Accuracy (%)**
Random Forest (ours)	93.77
XG Boost (ours)	93.02
LSTM (ours)	92.48
GRU (ours)	92.50
KNN (Benchmark)	98.00

Model	Accuracy (%)
MLP (Benchmark)	99.00
SVM (Benchmark)	99.00
RF (Benchmark)	99.00

The table above compares the validities of the models based on our study with the benchmarked accuracies in Ferriyan et al. study. Our models performed excellently, as is evident in the table. Nonetheless, the benchmark models performed better. The provision in the table snot only gives an achievement sensation but also provides an aspect to improve in case future models are to be developed. The comparison, in this case, shows the gap between our models and the benchmark. Particularly, the Random Forest had the least performance difference. The deep learning models exhibited superior performance; however, when compared with the KNN, MLP, and SVM, our models had a lower performance. In conclusion, all the models performed excellently in detecting intrusions in our case. The strengths and weaknesses of each model make them ideal for specific intrusions. The benchmark comparison, however, indicates that there is an opportunity to refine the models further. It is crucial to optimize the models to achieve utmost performance in this area of cybersecurity.

4.5 Discussion on Findings

After validating the models, we are left with an intricate network of findings that were both expected and unexpected. Much like any other organizational tool or framework, our models have proven to have its unique advantages and drawbacks, presenting the in-depth understanding and perspective. Therefore, in this section, we will discuss these while interpreting the findings from the comparison process.

4.5.1 Strengths and Weaknesses of Each Model

When it comes to the Random Forest, its main advantage is that it is naturally resistant to overfitting. Because it is an ensemble method, multiple decision trees are used to predict outcomes. If the trees are to some extent skewed, the overall model averages the values and avoids biases. Additionally, RF is well-suited to deal with missing data and will operate even if most data is lost. However, it is computationally demanding which is a downside for the HIKARI-2021 dataset, as previously stated. RF models are slow to train and slow to predict because of their multiple trees. Furthermore, RF might not be suited for specific tasks if there are nonlinear relationships between the variables. Since ID relies on such relationships, it might not always provide good performance.

XG Boost is one of the most extensively utilized libraries in machine learning and offers several advantages. It is much faster and performs better than SKlearn. Since it deals with missing values in a dataset as a matter of course, it has a greater capacity to handle missing data than any of the other models. One of the XG Boost's strengths is that it easily parallelizes and outperforms several other ensemble classifiers. XG Boost is preferred among other algorithms due to its sequential design. The constraint is the unsuitability for real-time forecasting situations. In the meantime, XG Boost tends to overfit noisy data sets, and it requires a lot of time to adjust hyper-parameters manually.

LSTMs are particularly well-suited for sequential data. Since they can store prior data in memory, they are particularly useful for time-series, such as network data. Second, the vanishing gradient problem that most conventional RNNs face is substantially reduced. LSTMs, on the other hand, are not without their drawbacks. First and foremost, they might become computationally expensive and take longer to train, especially with large amounts of data. Determining the finest architecture, as measured by the number of layers or components in layers, can also be challenging and need several iterations.

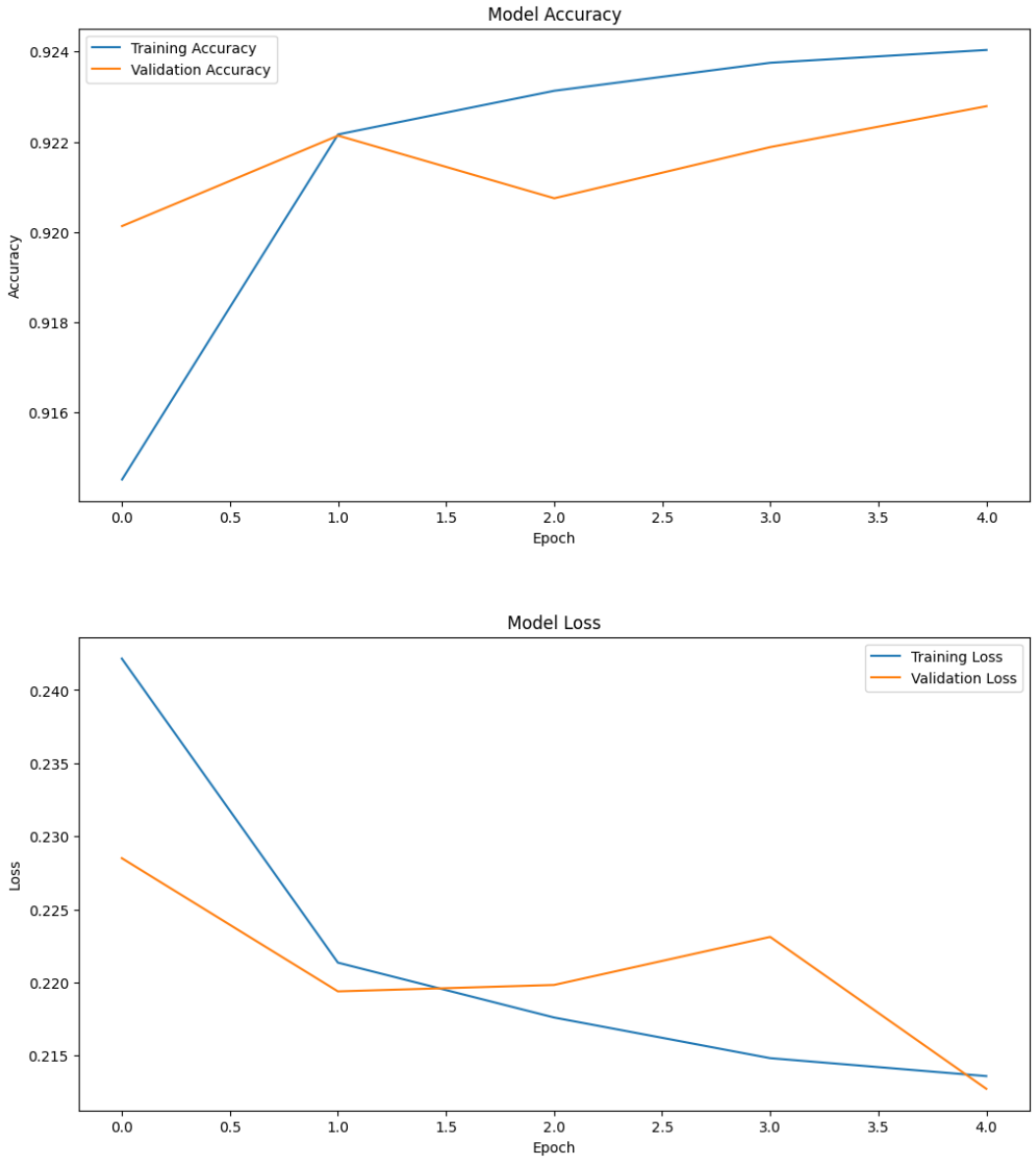

Figure 14. Model Accuracy and Loss for LSTM Algorithm Implemented

GRUs were similarly created as another option and offered similar benefits as LSTMs yet are less effective and speedier because of a less difficult design. This makes them less attractive compared to LSTMs. While this is undeniably obvious, especially with less complex datasets or tasks, it is much less certain with extraordinarily perplexing information and activities. Compared to LSTMs, GRUs cannot catch long-term dependencies unless more hidden layers are added, and more parameters are utilized.

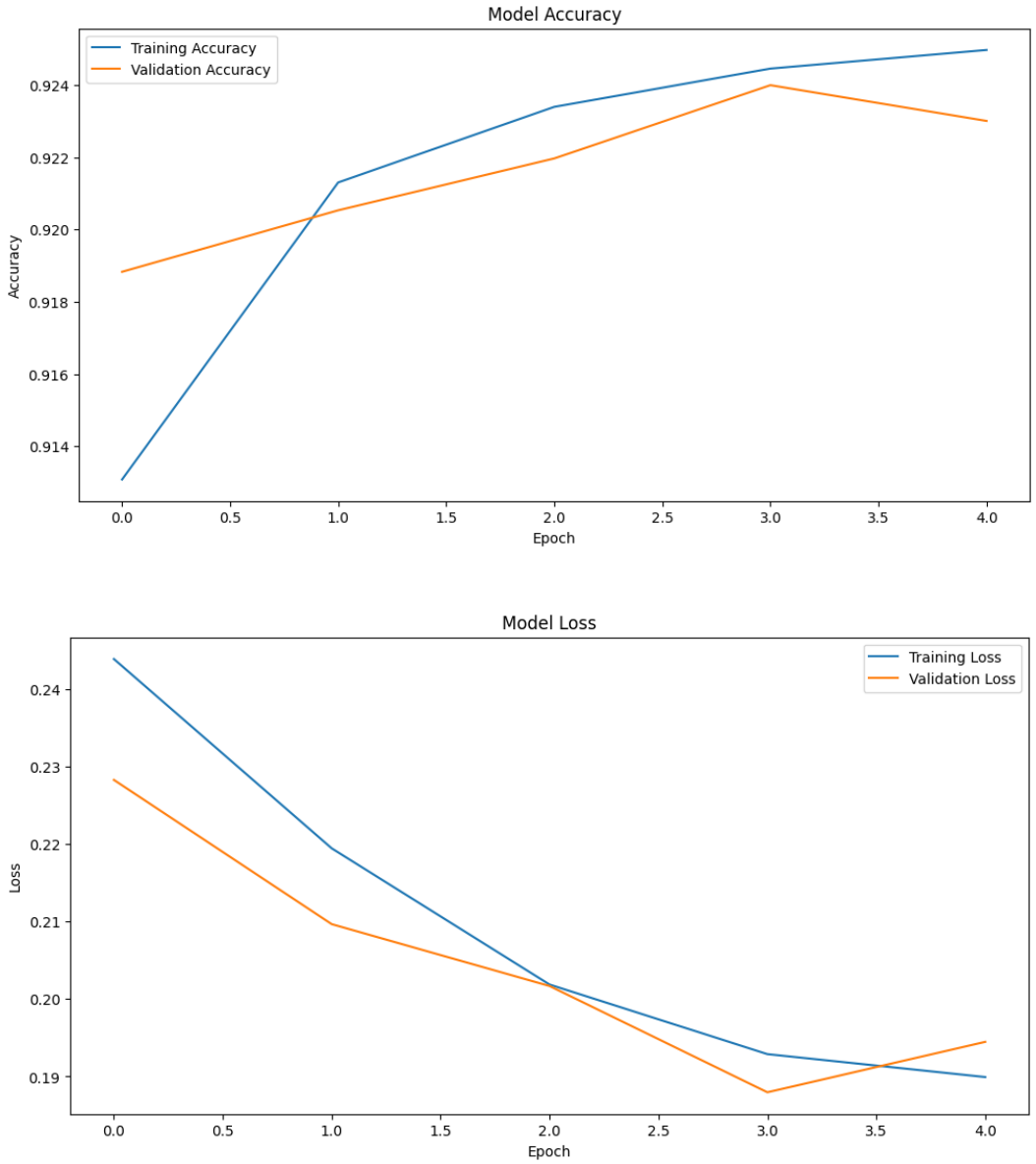

Figure 15. Accuracy and Error Rates for Deployed GRU Algorithm

4.5.2 Findings from Comparative Review

Primarily, the comparative review reveals different dimensions. Our models, competent in their own right, exhibit small accuracy differences with the benchmark. This could be due to various reasons, including but not limited to different datasets, different feature engineering methods, and/or the nature in which the models themselves operate. Benchmark models, especially the SVM and MLP, have marginally higher accuracies. However, as stated, a high accuracy does

not necessarily mean that this model is robust across datasets, or even against new data.

By comparing the models in question, the ensemble models, like Random Forest and XG Boost, are slightly more effective compared to the deep learning models, a pattern worth thinking about. This could mean that the ensemble models are more effective in generalizing this context, or the deep learning models require a more thorough architectural check or different hyperparameter tuning.

4.6 Challenges Experienced During Experimental Procedures

The path of experimentation was difficult. To start with, the HIKARI-2021 data set is too large and intricate for models to process quickly, offering reliable predictions. Thus, trying to ensure stable memory and reasonable training time for models was challenging. As an example, internal testing of hyperparameters for models such as XG Boost and LSTM took an enormous amount of time for the author. It was difficult to understand the optimal correlation of overfitting and underfitting, including complex dependencies like dropouts, the number of trees or layers, and the speed of learning in the forest of various determining entities.

Next, there is an issue with using the models in real-time; some models are useless in real-time predictions. For instance, in the case of intrusion detection, no model can be utilized without the other one acting as an immediate system if it cannot get the danger coming. However, during the process of building the architecture of LSTM and GRU in deep learning, multiple iterations were required. Again, the issues were not only the working solution but also what makes one architecture better than other. In summary, even though there were problems trouble, they all helped to understand the subject to give suggestions for more IDS experimentation.

4.7 Consequences of the Findings

Overall, the outcomes of the experiment of the study have remarkable implications, especially in the intrusion detection system field. There are several aspects in which the implications can be considered: the feasibility of deploying such models in the real world, the scalability of the experiment systems, and the relative effectiveness in identifying and reducing the threats.

In the context of the Random Forest and XGBoost models, which demonstrated high accuracy, it can be said that they are very good at identifying network intrusions. Therefore, ensemble methods, that is, those that integrate the results of multiple algorithms to make a prediction, are probably better at accommodating the variability and fluctuations of cyberattack patterns. Nevertheless, accuracy is not the only indicator of the model. Factors such as the computational expense and real-time response to a new query are also of paramount importance. Since Random Forest can be computationally demanding at times, especially with large datasets like HIKARI-2021, it may be difficult to apply it in many real environments where the computational apparatus is scarce.

XG Boost, offers promising potential for both high efficacy and efficiency. Its relatively high accuracy and superior computational efficiency could make it an ideal candidate for real-time intrusion detection – especially in applications where a high level of dynamic, emerging threats require immediate action.

Other implications, such as data science models based on deep learning like GRU and LSTM. However, deep learning models still have much space for further exploration as they still do not show the best results in terms of accuracy in comparison with ensemble models in the present study. Not only are they able to identify subtle patterns in large data sets more easily, but they have also been simpler to manage in sequential data. This could be of significant interest when it comes to rapidly spreading strains of sophisticated attacks that are difficult to

identify by ordinary algorithms. The only concern is how to optimize training and architecture configurations to maximize efficiency.

Meanwhile, the ratio of false positives to false negatives can bring another potential implication. While the high rate of false positives can only make staff feel frustrated or waste their time by triggering unnecessary alerts to the trigger point teams, the significantly high rate of false negatives can threaten the system due to the possibilities of the need to escalate real risks too late. Therefore, one more way to look at various models is to compare their performance in terms of this aspect.

4.8 Summary of Key Findings

Closing the chapter, the following summarizes the extensive experimental results and discoveries in the research. In this research, terrible informed the world of intrusion detection experienced the most potential findings from which competent models can be generated for the job. The Random Forest and XG boost performed well for this task. This was clear evidence that the ensemble method plays a role in actualizing a strong dominance against the intricate nature of identifying intrusion. The nature of the ensemble method to combine several algorithms to produce an aggregated result is a strong indicator of their potential, mainly when used in a generalized intrusion pattern. It appears they excellently focus on the mandate of generalization.

While LSTM and GRU might not be the standout performers from this particular experiment, but their performance deserves recognition. The outcomes indicate the significant wellspring of capacities lying within those models waiting to be tapped and refined for the task. Through the necessary adjustments and possibly more detailed, nuanced data, these models will likely set new performance standards in any field of intrusion detection. Apart from the mere accuracy, the model's performance in this study highlighted other vital aspects of the model's operationalization. The cost of execution, scalability, and ability to provide real-

time predictions emerged as critical factors. Whereas XG Boost, for example, appeared to provide a balance between accuracy and computational efficiency, other models, such as Random Forest, came with significant scalability challenges. The need for high accuracy not just about detecting more malicious activities but also doing so without false alarms and failed detections. The trade-off between false alarm rates and overlooks is crucial and could directly influence the models' operationalization in real-world applications. The need to get the correct detection right without too many alarms remains a significant challenge. The models generated, when compared to past studies yields competitive results. While there could be more room for improvement, compared to the 99% performance, it signals a way forward for future iterations and optimization. Essentially, this research in its intricate experimentation and assessment lays the foundation for the future of intrusion detection. It is a critical path, riddled with dead ends, empowerment, and rebirths, that points to what future research in critical cybersecurity may look like.

5 CONCLUSIONS

The aim of this thesis was to engage in deep research in the ML and DL methodology domain and investigate their ability to overcome a modern challenge of network intrusion detection. With the ever-growing river of existing and novel cyber threats, the importance of developing smarter, more performant, and adaptable models to foresee and prevent these threats is fundamental. I have also been able to understand all the details, advantages, as well as the most common limitations of the various ML and DL models that were involved in completing this assignment. At the beginning of the research, the hypothesis was formulated so that under the correct arrangement and assessment, some algorithms of ML and DL could achieve greater results in detecting network intrusions than the common method. To verify this, XG Boost, Random Forest, LSTM, and GRU were selected as the top 4 models. These models were chosen because their theoretical underpinnings are extensively studied. Even though tree-based models have a substantial benefit in high-performance classification, deep learning mechanisms, specifically LSTM and GRU, can recognize intricate patterns in serial data required for our network packet.

In this project, our target was determined based on the HIKARI-2021 dataset, where we conducted an initial exploration of the data to familiarize with the target dataset. This helped us understand the distribution, statistical attributes, and distribution of attack classes in the data. We made decisions regarding the pre-processing steps based on the knowledge we obtained from this exploration. As discussed above, although each model had its own strengths and weaknesses, for example: The Random Forest model had the high level of accuracy, 93.77%, showing the applicability of ensemble learning in handling large datasets with many different features. The XG Boost model, a tree-based method, achieved an accuracy of 93.02%, with the same explanation as the Random Forest model assert that gradient boosting helps enhance the decision-making process in multiple ways. From a DL perspective, the LSTM model had an accuracy of 92.48% and the

GRU model had an accuracy of 92.50%, both with a slightly low accuracy rate than others, as deep learning models can be highly sensitive and complex, especially when used for non-sequential features. However, the differences were quite close, and it was interesting to notice that LSTM and GRU performed almost as well. On the other hand, the models obtained did not quite reach the level of models such as KNN, MLP, and SVM, capable of more than 98-99% accuracy. The results obtained show that the field of the creation of idealized intrusion detection systems still requires research.

The results obtained have several implications. Firstly, they suggest that machine learning and deep learning ML and DL are suitable for detecting intrusions. While almost 95% accuracy is not as high as the aforementioned models, it is still relatively high, meaning that the models are viable for actual implementation. At the same, however, the results we obtained also show the necessity of choosing models. The fact that GRU performed slightly better than LSTM, while Conv1D was the least accurate, shows that for different data and tasks, different models may be more or less effective. Hence, further comparisons and future individual model testing can be useful for uncovering the effectiveness of different model's.

In conclusion, this thesis represented a well-structured introduction to the network intrusion detection field through a mixture of ML and DL. The results were promising but managed to identify where future research and optimization should be applied. Understanding the functioning principles, the bright sides, but also the possible weaknesses of each of them, this work creates a base for a better and more concise future regarding the identification of cyberattacks. This area, due to problems presented, should constantly evolve, and so should the countermeasures, and this thesis managed to create a significant first step in that process.

The outcome of the research has resulted in many opportunities and prospects to do more work around Network Intrusion Detection in the future. As prospects, we

have the opportunity to engage in the improvement, development, and expansion of some aspects, the study of other vectors for even better security measures.

Even though the models used in this study performed well, there is always the possibility of achieving even higher accuracy. First, more complicated optimization methods may achieve better results. For example, hyperparameter tuning using grid search or random search, or an ensemble method, which mixes several models to increase the accuracy. Second, experimenting with model architectures, particularly works well with deep learning models, may lead to better outcomes.

The field of machine learning, and deep learning in particular, is evolving rapidly, and new models and methods keep emerging. For instance, the Transformer-based models, such as BERT and its derivatives, have demonstrated excellent results in a variety of NLP tasks. Perhaps they may offer a distinctive contribution to intrusion detection techniques, making the detection process even more reliable and accurate.

The performance of an intrusion detection model depends significantly on the attributes upon which the model is trained. Future research can focus more on the feature selection techniques, finding a better portion of the attributes that provide the ideal functionality of the model. Moreover, the development of more features, which contain complex attributes of network packets, can also be subject to research.

It is important to note that a completely static dataset was analysed. In practice, network intrusion detection operates dynamically and in real-time. It is possible to verify the practicality and reality-response of the models by testing them in such a situation.

While encryption is increasingly common in a privacy-oriented digital world, encrypted traffic can be a major hurdle. Future research may explore the issues

and complexities surrounding detection on encrypted data, ensuring that protective measures are effective even if the data is weakened.

Rather than solely relying on single models, there is a possibility for the development of holistic systems that include several models, heuristics, and rules. This system can complement its individual constituents by using them at the optimal times, allowing the system to identify an intrusion with the fewest false positives and the best accuracy.

As detection systems become more advanced, so do the attacking strategies. Adversarial attacks on deep learning models are increasing in prevalence. Students should research these adversarial strategies and subsequently develop defenses to address these potential threats towards the intrusion detection system.

In numerous real-world scenarios, as soon as the intrusion was noticed, it was given to human for verification. This verification is the subject of feedback to the model. The systems of the future will also have to develop feedback loops where the model, from feedback given by human experts, learns continuously so that the current model is even more accurate.

To conclude, although this thesis has already covered a long way in work of network intrusion detection, there are still a lot of exciting challenges ahead. The fields of machine learning and cybersecurity continuously influence each other's development and evolution in an endless cycle of new approaches to attack and defend. Hopefully, the future research will focus on developing new approaches and tools based on the results of this thesis, encouraging, and boosting innovation, and securing the digital realms from the new threats as they appear.

REFERENCES

[1] Alferidah, D. K., & Jhanjhi, N. Z. (2020, October). Cybersecurity impact over bigdata and iot growth. In 2020 International Conference on Computational Intelligence (ICCI) (pp. 103-108). IEEE.

[2] Yeboah-Ofori, A., Ismail, U. M., Swidurski, T., & Opoku-Boateng, F. (2021, July). Cyberattack ontology: A knowledge representation for cyber supply chain security. In 2021 International Conference on Computing, Computational Modelling and Applications (ICCMA) (pp. 65-70). IEEE.

[3] Xin, Y., Kong, L., Liu, Z., Chen, Y., Li, Y., Zhu, H., ... & Wang, C. (2018). Machine learning and deep learning methods for cybersecurity. IEEE access, 6, 35365-35381.

[4] Ferriyan, A., Thamrin, A. H., Takeda, K., & Murai, J. (2021). Generating network intrusion detection dataset based on real and encrypted synthetic attack traffic. Applied Sciences, 11(17), 7868.

[5] Velan, P., Čermák, M., Čeleda, P., & Drašar, M. (2015). A survey of methods for encrypted traffic classification and analysis. International Journal of Network Management, 25(5), 355-374.

[6] De Lucia, M. J., & Cotton, C. (2018, May). Identifying and detecting applications within TLS traffic. In Cyber Sensing 2018 (Vol. 10630, pp. 179-190). SPIE.

[7] Kaur, S., & Singh, M. (2013). Automatic attack signature generation systems: A review. IEEE Security & Privacy, 11(6), 54-61.

[8] Ahmed, M., Mahmood, A. N., & Hu, J. (2016). A survey of network anomaly detection techniques. Journal of Network and Computer Applications, 60, 19-31.

[9] Zeek IDS. 2021. Retrieved May 10, 2024, from https://zeek.org.

[10] Shiravi, A., Shiravi, H., Tavallaee, M., & Ghorbani, A. A. (2012). Toward developing a systematic approach to generate benchmark datasets for intrusion detection. computers & security, 31(3), 357-374.

[11] Moustafa, N., & Slay, J. (2016). The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. Information Security Journal: A Global Perspective, 25(1-3), 18-31.

[12] Maciá-Fernández, G., Camacho, J., Magán-Carrión, R., García-Teodoro, P., & Therón, R. (2018). UGR '16: A new dataset for the evaluation of cyclostationarity-based network IDSs. Computers & Security, 73, 411-424.

[13] Siddique, K., Akhtar, Z., Khan, F. A., & Kim, Y. (2019). KDD cup 99 data sets: A perspective on the role of data sets in network intrusion detection research. Computer, 52(2), 41-51.

[14] Özgür, A., & Erdem, H. (2016). A review of KDD99 dataset usage in intrusion detection and machine learning between 2010 and 2015.

[15] Luo, C., Wang, L., & Lu, H. (2018). Analysis of LSTM-RNN based on attack type of kdd-99 dataset. In Cloud Computing and Security: 4th International Conference, ICCCS 2018, Haikou, China, June 8-10, 2018, Revised Selected Papers, Part I 4 (pp. 326-333). Springer International Publishing.

[16] Fontugne, R., Borgnat, P., Abry, P., & Fukuda, K. (2010, November). Mawilab: combining diverse anomaly detectors for automated anomaly labeling and performance benchmarking. In Proceedings of the 6th International COnference (pp. 1-12).

[17] Hafsa, Mounir, and Farah Jemili. (2019). Comparative Study between Big Data Analysis Techniques in Intrusion Detection. Big Data and Cognitive Computing 3, no. 1: 1. Retrieved from https://doi.org/10.3390/bdcc3010001.

[18] Kim, J., Sim, C., & Choi, J. (2019, June). Generating labelled flow data from MAWILab traces for network intrusion detection. In Proceedings of the ACM Workshop on Systems and Network Telemetry and Analytics (pp. 45-48).

[19] CAIDA Datasets. 2021. Retrieved May 10, 2024, from https://www.caida.org/catalog/datasets/completed-datasets/.

[20] Jonker, M., King, A., Krupp, J., Rossow, C., Sperotto, A., & Dainotti, A. (2017, November). Millions of targets under attack: a macroscopic characterization of the DoS ecosystem. In Proceedings of the 2017 Internet Measurement Conference (pp. 100-113).

[21] Lutscher, P. M., Weidmann, N. B., Roberts, M. E., Jonker, M., King, A., & Dainotti, A. (2020). At home and abroad: The use of denial-of-service attacks during elections in nondemocratic regimes. Journal of Conflict Resolution, 64(2-3), 373-401.

[22] Sabahi, F., & Movaghar, A. (2008, October). Intrusion detection: A survey. In 2008 Third International Conference on Systems and Networks Communications (pp. 23-26). IEEE.

[23] N'Cir, C. E. B., Cleuziou, G., & Essoussi, N. (2015). Overview of overlapping partitional clustering methods. Partitional Clustering Algorithms, 245-275.

[24] Al-Yaseen, W. L., Othman, Z. A., & Nazri, M. Z. A. (2017). Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system. Expert Systems with Applications, 67, 296-303.

[25] Le, T. M., Vo, T. M., Pham, T. N., & Dao, S. V. T. (2020). A novel wrapper–based feature selection for early diabetes prediction enhanced with a metaheuristic. IEEE Access, 9, 7869-7884.

[26] Bharati, M., & Tamane, S. (2017, October). Intrusion detection systems (IDS) & future challenges in cloud-based environment. In 2017 1st International Conference on Intelligent Systems and Information Management (ICISIM) (pp. 240-250). IEEE.

[27] El-Desoky, A. E., Ali, H. A., & Azab, A. A. (2007, November). A pure peer-to-peer desktop grid framework with efficient fault tolerance. In 2007 International Conference on Computer Engineering & Systems (pp. 346-352). IEEE.

[28] Azab, A. A., & Kholidy, H. A. (2008, November). An adaptive decentralized scheduling mechanism for peer-to-peer desktop grids. In 2008 International Conference on Computer Engineering & Systems (pp. 364-371). IEEE.

[29] Bace, R., & Mell, P. (2001). NIST special publication on intrusion detection systems. National Institute of Standards and Technology, 16.

[30] Debar, H., Dacier, M., & Wespi, A. (1999). Towards a scientific classification of interruption recognition frameworks. Computer Networks, 31(8), 805-822.