Journal of Network and Information Security
13 (1) 2025, 01-17
www.publishingindia.com/jnis

Using Machine Learning to Detect Unusual Patterns and Behaviors in Network Security

Hameeda Abubakar Aminu[1] and Jesse Mazadu Ismaila[2*]
[1]Computer Science Department, Federal University Wukari, Nigeria.
Email: hameejal07@gmail.com
[2]Computer Science Department, Federal University Wukari, Nigeria.
Email: jesse@fuwukari.edu.ng
[*]Corresponding Author

Abstract: The development of the network anomaly detection model leveraged the computational robustness of the Anaconda programming environment. The concluding phase of this study, classification and performance evaluation, leverages features selected through the LDA algorithm to train machine learning classification models. Algorithms such as Random Forest and Support Vector Machine are employed, and the individual models undergo rigorous evaluation using metrics such as accuracy, precision, recall, and F1-score. This meticulous evaluation process ensures the reliability and efficacy of the models in discerning network intrusions. The Support Vector Machine (SVM) classification report for network anomaly detection using the CIC-DDoS2019 dataset indicates high performance across various evaluation metrics. With a precision of 0.9966 for class 0 and 0.9973 for class 1, the SVM demonstrates a strong ability to accurately classify instances of both normal and anomalous network traffic. Similarly, the recall scores of 0.9973 for class 0 and 0.9965 for class 1 indicate the model's effectiveness in identifying the majority of instances belonging to each class. The F1 scores, which consider both precision and recall, are also high for both classes, with values of 0.9970 for class 0 and 0.9969 for class 1. Moreover, the overall accuracy of the SVM model is reported as 0.9969, indicating its ability to correctly classify a large proportion of instances in the dataset. By delineating the research methodology into these phases, leveraging advanced computational tools and libraries, and reporting high-performance metrics for the SVM model, this study presents a comprehensive framework for developing an effective network anomaly detection system using machine learning algorithm.

Keywords: Anomaly, Detection, Intrusion, Network, Support vector machine.

I. Introduction

Background of the Study

The advent of Internet technology has ushered in an era of remarkable technological progress, yet alongside these advancements; it has also introduced new avenues for network intrusion and privacy breaches [1]. Consequently, anomaly detection has emerged as a critical endeavour across diverse fields, drawing upon centuries of research and development. Over time, a myriad of methodologies has been devised and deployed to address this challenge, spanning various domains and applications. At its essence, anomaly detection entails the identification of patterns within data that diverge from expected norms, as outlined by [2]. The significance of anomaly detection is

profound and far-reaching, given the inherent risks associated with undetected anomalies, which often carry pivotal and actionable insights. For instance, anomalies in computer network traffic may signify a cyber-attack stemming from a compromised system, while anomalies in credit card transaction data could indicate fraudulent activities such as theft.

Surveys conducted have revealed that firewalls, Deep Packet Inspection (DPI) systems and Intrusion Detection Systems (IDS) are the typical methods for anomaly detection, however, the cost to deploy these countermeasures and the complexity of the system have to be considered [3]. Hence, several researchers have applied the viabilities of Machine Learning (ML) algorithms to security in various types of network anomalies, from the traditional computer network to the IoT network [4]. However, the dataset employed for anomaly analysis is extensive, resulting in a notable decline in the efficiency of the machine learning algorithm. Authors have commonly resorted to feature selection methods to address the challenge of dataset dimensionality. A. Bemmert *et al.* [5] noted that feature selection involves reducing high-dimensional datasets to retain only the most relevant features for analysis.

The efficacy of machine learning (ML) techniques, coupled with a feature selection strategy, has been promising, leading to the proposal of employing Support Vector Machine (SVM) and Random Forest (RF) algorithms on the Canadian Institute for Cybersecurity - Distributed Denial of Service 2019 (CIC-DDoS2019) dataset sourced from the Mendeley machine learning repository. A factual reason is that upon examining the dataset, it became evident that its abundance of features could potentially hinder the efficiency of the models. Consequently, the study advocates for the implementation of the Linear Discriminant Algorithm (LDA) for feature selection. Thus, a significant contribution of this research lies in leveraging the LDA algorithm to select features before applying the SVM and RF algorithms for further feature selection.The rapid evolution of network technology has led to a proliferation of network services and applications, providing hackers with expanded avenues to compromise network security. Of particular concern is the paradigm shift in operational frameworks between next-generation networks and legacy systems, posing new challenges for anomaly detection methods [6]. To address these evolving threats and adapt to dynamic network environments, there is a pressing need for the advancement of anomaly detection techniques. Consequently, this study aims to explore the efficacy of Support Vector Machine (SVM) and Random Forest (RF) machine learning algorithms, augmented by feature selection using the linear discriminant algorithm, in mitigating emerging security threats within modern network infrastructures. The aim of this study is to develop a network anomaly detection system using machine learning algorithms. The objectives include:

- Perform feature selection using the LDA algorithm.
- Apply the viabilities of the SVM and RF algorithm for the detection of network anomaly.
- Evaluate the performance of the models using the accuracy, precision and recall metrics.

The significance of this study is multifaceted and crucial in the realm of cybersecurity. The escalating sophistication of cyber threats, poses an urgent demand for robust network anomaly detection systems to bolster defence mechanisms against potential breaches. By leveraging the SVM and RF machine learning algorithms and feature selection techniques, this study aims to elevate the accuracy and efficacy of anomaly detection processes. Such enhancements are pivotal in fortifying network security measures and preemptively identifying anomalous activities that may indicate malicious intent or cyber-attacks. Moreover, the outcomes of this research hold practical implications for various stakeholders, including network administrators, cybersecurity professionals, and organizations at large. Armed with insights gleaned from this study, these entities can make informed decisions and implement proactive measures to safeguard their networks, assets, and sensitive data from emerging cybersecurity threats. Thus, the significance of this study lies in its potential to contribute to the ongoing efforts to fortify cybersecurity defences and

mitigate the risks posed by network anomalies and cyber-attacks.

II. Literature Review

A. Theoretical Review

The rapid growth of computer networks has enabled them to function as a central information system in modern life. The increase in the size, services and applications, and infrastructure of computer networks such as the Internet of Things, has made them complex and heterogeneous. Thus, they confront various critical threats such as malicious activities, network intruders and cybercriminals. Identifying and preventing these detrimental cyber activities are a high priority these days [7]. Analyzing and monitoring network traffic to identify such malicious actions in large-scale networks are crucial tasks, and ideally should be carried out automatically with little supervision by network administrators [8]. Anomaly detection is a data analysis task where the goal is to detect patterns deviating greatly from normal data. It is suitable for automatically identifying illegal, malicious activities and other forms of network abuse from the normal behaviors of network systems. Anomalies pose a problem in various application areas, such as manufacturing, medical or communication systems. They often lead to a decrease in system performance and can cause instabilities and failure. Often, the cause of anomalies is unknown effects within complex systems. Therefore, the capability of understanding and detecting these underlying effects with the aid of data is the key to ensuring the desired outcome of complex technical systems [9].

B. Feature Selection

Feature selection is a technique that reduces dimensionality by retaining only relevant qualities and eliminating unnecessary and redundant ones [10]. Reducing the dimensionality of input can improve performance in two ways: it can improve generalization and classification accuracy but also slow learning and increase model complexity. Selecting the right characteristics can improve problem comprehension and reduce measurement costs. In some cases, feature selection can make a big difference. For example, only two characteristics out of 7129 can be employed to improve classification performance in microarray data processing. Reducing the dimensionality of problems while reducing related costs is the aim of feature selection in machine learning, and deep learning methodologies. Such applications include, for example, deriving information from photographs and interpreting expert differences in illness diagnosis. Filter, wrapper, and embedding approaches are some of the feature selection strategies [11].

C. Review of Related Works

Related works on anomaly detection are presented according to machine learning approaches and deep learning approaches thus:

i) Machine Learning Approaches

M. Nicolau, and J. McDermott [12] implemented a learning neural representation for network anomaly detection. Their paper proposed a latent representation model for improving network anomaly detection. Well-known anomaly detection algorithms often suffer from challenges posed by network data, such as high dimension and sparsity, and a lack of anomaly data for training, model selection, and hyperparameter tuning. Their approach was to introduce new regularizes to a classical autoencoder (AE) and a variational AE, which force normal data into a very tight area centered at the origin in the nonsaturating area of the bottleneck unit activations. These trained AEs on normal data will push normal points toward the origin, whereas anomalies, which differ from normal data, will be put far away from the normal region. The models were very different from common regularized AEs, sparse AEs, and contractive AEs, in which the regularized AEs tend to make their latent representation less sensitive to changes in the input data. The bottleneck feature space was now used as a new data representation. Several one-class learning algorithms were used for evaluating the proposed models. The experiments revealed that their models helped these classifiers to perform efficiently and consistently on high-

dimensional and sparse network datasets, even with relatively few training points. More importantly, the models can minimize the effect of model selection on these classifiers since their performance was insensitive to a wide range of hyperparameter settings.

D. Kwon *et al*. [13] performed an empirical study on network anomaly detection using convolutional neural networks. In their study, they empirically evaluated a set of deep learning models, including Fully Connected Network (FCN), Variational Auto Encoder (VAE), and Sequence to Sequence model with Long Short-Term Memory (Seq2Seq-LSTM), for network anomaly detection. In addition, they further evaluated Convolution Neural Networks (CNNs) for network anomaly detection in this study. They set up three simple CNN models with different internal depths (shallow CNN, moderate CNN, and deep CNN) to see the impact of the depth on the performance. They now evaluated the models using three different types of traffic datasets. Their experimental results show that deeper structures do not make any performance improvement. In addition, they observed that the evaluated CNN models occasionally outperform the VAE models, but do not work better than the other deep learning models based on FCN and Seq2Seq-LSTM.

B. J. Radford *et al.* [14] in their work "Network traffic anomaly detection using recurrent neural networks" showed that a recurrent neural network can learn a model to represent sequences of communications between computers on a network and can be used to identify outlier network traffic. Defending computer networks is a challenging problem and is typically addressed by manually identifying known malicious actor behaviour and then specifying rules to recognize such behaviour in network communications. However, these rule-based approaches often generalize poorly and identify only those patterns that are already known to researchers. An alternative approach that does not rely on known malicious behaviour patterns can potentially also detect previously unseen patterns. They tokenize and compress netflow into sequences of "words" that form "sentences" representative of a conversation between computers. These sentences were then used to generate a model that learns the semantic and syntactic grammar of the newly generated language. They use Long-Short-Term Memory (LSTM) cell Recurrent Neural Networks (RNN) to capture the complex relationships and nuances of this language. The language model was then used to predict the communications between two IPs and the prediction error was used as a measurement of how typical was the observed communication. By learning a model that was specific to each network, yet generalized to typical computer-to-computer traffic within and outside the network, a language model can identify sequences of network activity that are outliers concerning the model. They demonstrated positive unsupervised attack identification performance (AUC 0.84) on the ISCX IDS dataset which contains seven days of network activity with normal traffic and four distinct attack patterns.

G. Fernandes *et al.* [15] carried out a comprehensive survey on network anomaly detection. Their objective for this study was to review the most important aspects of anomaly detection, covering an overview of a background analysis as well as a core study on the most relevant techniques, methods, and systems within the area. Therefore, to ease the understanding of this survey's structure, the anomaly detection domain was reviewed under five dimensions: (1) network traffic anomalies, (2) network data types, (3) intrusion detection systems categories, (4) detection methods and systems, and (5) open issues. The paper concluded with an open issues summary discussing presently unsolved problems, and final remarks.

A. Nagaraja *et al.* [16] worked on similarity-based feature transformation for network anomaly detection where feature reduction was achieved using the proposed feature transformation technique. However, their approach for feature transformation used the proposed Gaussian distance function to achieve dimensionality reduction to represent the original input dataset in the new transformation space. They further proposed a new computation expression for determining equivalent deviation and threshold in Gaussian space. Experiments were performed on KDD and NSL-KDD datasets by considering widely applied classifier algorithms in various state-of-the-art research contributions. For

performance validation of machine learning models, k-fold cross-validation was applied by setting k to 10 by considering evaluation parameters such as accuracy, precision and recall. Experiment results proved that their approach for anomaly detection that applies the proposed feature transformation technique proved comparatively better than detection methods CANN, GARUDA, and UTTAMA addressed in the recent research literature.

ii) Deep Learning Approaches

S. Naseer *et al*. [17] in their work "Enhanced network anomaly detection based on deep neural networks" investigated the suitability of deep learning approaches for anomaly-based intrusion detection systems. For this research, they developed anomaly detection models based on different deep neural network structures, including convolutional neural networks, autoencoders, and recurrent neural networks. These deep models were trained on NSLKDD training data set and evaluated on both test datasets provided by NSLKDD, namely NSLKDDTest+ and NSLKDDTest21. All experiments in this paper were performed by authors on a GPU-based test bed. Conventional machine learning-based intrusion detection models were implemented using well-known classification techniques, including extreme learning machine, nearest neighbor, decision tree, random forest, support vector machine, naive bays, and quadratic discriminant analysis. Both deep and conventional machine learning models were evaluated using well-known classification metrics, including receiver operating characteristics, the area under the curve, precision-recall curve, mean average precision and accuracy of classification. Experimental results of deep IDS models showed promising results for real-world applications in anomaly detection systems.

A. H. Hamamoto *et al.* [18] in their research "Network anomaly detection system using genetic algorithm and fuzzy logic" developed a scheme combining Genetic Algorithm and Fuzzy Logic for network anomaly detection. The Genetic Algorithm was used to generate a Digital Signature of Network Segment using Flow Analysis, where information extracted from network flow data was used to predict the network traffic behavior for a given time interval. Furthermore, a Fuzzy Logic scheme was applied to decide whether an instance represents an anomaly or not, differing from some approaches present in the literature. Indeed, it was proposed an expert system with the capability to monitor the network's traffic with IP flows while expected behaviors are generated on a regular time interval basis, issuing alarms when a possible problem is present. The proposed anomaly detection system exposed network problems autonomously. The results acquired from applying the proposed approach in real network traffic flows achieve an accuracy of 96.53% and a false positive rate of 0.56%. Moreover, their method succeeded in achieving higher performance compared to several other approaches.

D. Kwon *et al*. [19] embarked on a survey of deep learning-based network anomaly detection. They presented an overview of deep learning methodologies, including restricted Bolzmann machine-based deep belief network, deep neural network, and recurrent neural network, as well as the machine learning techniques relevant to network anomaly detection. In addition, this article introduced the latest work that employed deep learning techniques with a focus on network anomaly detection through the extensive literature survey. They also discussed their local experiments showing the feasibility of the deep learning approach to network traffic analysis. M. Said *et al.* [20] proposed a network anomaly detection using LSTM based autoencoder. They proposed a hyper approach based on Long Short Term Memory (LSTM) autoencoder and One-class Support Vector Machine (OC-SVM) to detect anomaly-based attacks in an unbalanced dataset, by training the models using only examples of normal classes. The LSTM-autoencoder was trained to learn the normal traffic pattern and to learn the compressed representation of the input data (i.e. latent features) and then fed it to an OC-SVM approach. The hybrid model overcomes the shortcomings of the separate OC-SVM, which is its low capability to operate

with massive and high-dimensional datasets. Additionally, they performed their experiments using the most recent dataset (InSDN) of Intrusion Detection Systems (IDSs) for SDN environments. The experimental results showed that the proposed model provides a higher detection rate and reduces the processing time significantly. Hence, their method provided great confidence in securing SDN networks from malicious traffic.

B. Lindemann *et al*. [9] carried out a survey on anomaly detection for technical systems using LSTM networks. In their article, a survey on state-of-the-art anomaly detection was done using deep neural and especially long short-term memory networks. The investigated approaches were evaluated based on the application scenario, data and anomaly types as well as further metrics. To highlight the potential of upcoming anomaly detection techniques, graph-based and transfer learning approaches were also included in the survey, enabling the analysis of heterogeneous data as well as compensating for its shortage and improving the handling of dynamic processes. W. Xu *et al.* [21] worked on improving the performance of autoencoder-based network anomaly detection on NSL-KDD dataset. They proposed a novel 5-layer autoencoder (AE)-based model better suited for network anomaly detection tasks. Their proposal was based on the results they obtained through an extensive and rigorous investigation of several performance indicators involved in an AE model. In their proposed model, they used a new data pre-processing methodology that transforms and removes the most affected outliers from the input samples to reduce model bias caused by data imbalance across different data types in the feature set. The proposed model utilizes the most effective reconstruction error function which plays an essential role in the model to decide whether a network traffic sample is normal or anomalous. These sets of innovative approaches and the optimal model architecture allow their model to be better equipped for feature learning and dimension reduction thus producing better detection accuracy as well as F1-score. They evaluated their proposed model on the NSL-KDD dataset which outperformed other similar methods by achieving the highest accuracy and F1-score at 90.61% and 92.26% respectively in detection.

III. Methodology

In the pursuit of any scholarly investigation, the establishment of a robust research methodology serves as a fundamental pillar, orchestrating a systematic approach throughout the developmental trajectory. It delineates a structured pathway encompassing distinct phases, thereby shaping the proposed methodological framework. Within the specific purview of this study, the overarching aim is to devise a resilient and efficacious system for the detection of network intrusions. The envisaged research methodology is delineated into three principal phases: data preprocessing, feature selection, and classification and performance evaluation. The initial phase, data preprocessing, meticulously orchestrates the preparation and refinement of raw data to facilitate effective analysis. This involves intricate tasks such as handling missing values, normalizing data, and eliminating noise or outliers, with the primary objective of ensuring that the data is rendered in a format conducive to subsequent analysis.

Advancing further, the feature selection phase entails the discernment and extraction of pivotal attributes from the preprocessed data. The goal is to retain only those features wielding significant influence in the detection of network intrusions. To this end, the Linear Discriminant algorithm (LDA) is proposed, as it plays a crucial role in reducing dimensionality, enhancing model performance, and augmenting the interpretability of results. The concluding phase, classification and performance evaluation, leverages the features selected through the LDA algorithm to train machine learning classification models. Employing algorithms such as Random Forest and Support Vector Machine, the individual models undergo rigorous evaluation using metrics such as accuracy, precision, recall, and F1-score. This meticulous evaluation process ensures the reliability and efficacy of the models in discerning network intrusions. A visual representation of the proposed methodology is encapsulated in Fig. 1, providing a comprehensive overview of the sequential progression through the outlined phases. This methodological framework is structured not only to address the intricacies of network intrusion detection

but also to contribute valuable insights to the broader domain of cybersecurity research.

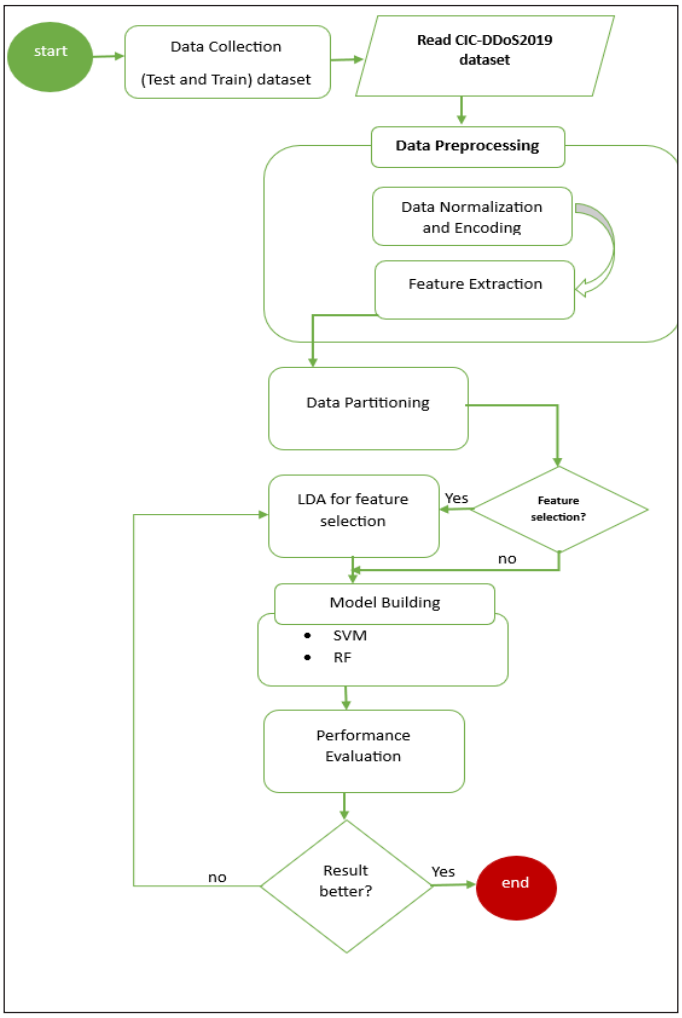

Fig. 1: The Model Flow Chart

A. Dataset Description

The dataset utilized to analyze network anomalies is the CIC-DDoS2019 dataset, provided by the Canadian Institute for Cybersecurity and obtained from the Mendeley machine learning repository. CIC-DDoS2019 comprises both benign and common DDoS attacks, making it representative of real-world network traffic (PCAPs). It includes labelled flows generated by CICFlowMeter-V3, indicating timestamps, source and destination IPs, ports, protocols, and attack types (stored in CSV files). The data was collected daily, with each day featuring raw network traffic (PCAPs) and event logs (Windows and Ubuntu event logs) per machine. The authors employed CICFlowMeter-V3 to extract over 80 traffic features from the raw data, saving them as CSV files per machine. The dataset encompasses various modern reflective DDoS attacks, such as PortMap, NetBIOS, LDAP, MSSQL, UDP, UDP-Lag, SYN, NTP, DNS, and SNMP.

B. Data Preprocessing

In the proposed dataset, it was observed that certain attributes exhibit nominal values. Hence, it becomes imperative to convert these attributes into numerical values during both the training and testing phases. Algorithm 1, delineates the procedure for this conversion of nominal attributes into numerical values. Initially, the algorithm calculates the total count of nominal features. Subsequently, for each nominal feature, details such as the feature name, the number of variables (k), and the frequency of each variable within the feature are specified. Following this, the recurring variables are organized in descending order based on their occurrence frequency, and each variable is then assigned a numerical value ranging from zero to K.

Algorithm 1: Nominal to Numeric Algorithm
Step 1: nominal feature counter: *c*
Step 2: set $i = 1$
Step 3: while $i!$ = c **then**
Step 4: k_i = *num of variable (i) involved in feature C,* p_i*: num of repeated variable (i) repeated*
Step 5: Order repeated variable from maximum to minimum number
Step 6: allocate the number from zero to *k* that was obtained from step 2 to variables
Step 7: $i = i + 1$ goto step 3
Step 1: stop

C. Feature Selection

Feature selection is a methodology utilized to decrease the dimensionality of a feature set by eliminating extraneous features, thus preserving only the most salient ones. In the realm of DDoS detection, feature selection is implemented to optimize the efficiency of the detection process. The

aim is to discern and retain the most pertinent features while discarding those that do not substantially contribute to the detection task. Consequently, to identify the relevant features, the study advocated for the utilization of an artificial intelligence technique, specifically the LDA algorithm. Linear Discriminant Analysis (LDA) is a statistical method primarily used for dimensionality reduction and classification. In the context of feature selection, LDA operates by maximizing the separation between multiple classes or categories within the data [22]. It achieves this by projecting the feature space onto a lower-dimensional subspace, where the class separation is maximized [23]. Specifically, LDA identifies the directions (or linear combinations of features) that maximize the ratio of between-class variance to within-class variance. Features that contribute the most to this separation are retained, while those with less discriminatory power are discarded. By selecting features that effectively discriminate between classes, LDA facilitates the creation of a compact and informative feature set, which is essential for improving the performance of classification algorithms, such as in the case of network anomaly detection.

D. Classification Algorithm

The investigation into network anomaly detection utilizing the CIC-DDoS2019 dataset entails constructing and validating models using a supervised machine learning framework, particularly classification methods, given the dataset's classification challenge of network attacks. Primarily, the objective is to predict the class label of each sample based on the features present in the CIC-DDoS2019 dataset. Consequently, the study advocates for the utilization of various machine-learning techniques in model construction, including Random Forest (RF) and Support Vector Machine (SVM) algorithms.

E. SVM Algorithm

A Support Vector Machine (SVM) constructs hyperplanes in a high-dimensional space to facilitate classification, regression, or outlier detection. Its primary objective is to find an optimal linear hyperplane that maximizes the margin of separation between binary classes. This approach involves using a subset of the data to train the model, identifying support vectors that represent the training data, and utilizing these support vectors to classify unseen samples into target classes. In this study, an SVM model is developed to classify network anomalies as either normal or anomalous (indicative of an attack). When an intrusive connection occurs, the SVM model identifies the anomaly. The classification process involves using training and testing sets comprising instances of connections, where each instance includes a target value (normal or attack) and features identifying the entire instance [24]. Through SVM, this research constructs a model capable of predicting the presence or absence of a denial-of-service attack in the test set using a parameter configuration generated during model construction, training, and tuning. The SVM training algorithm, given a set of training samples belonging to two classes, builds a model that assigns new samples to one of the classes. It operates as a non-probabilistic binary classifier. By employing the kernel option, an SVM model can perform non-linear classification, albeit with a careful selection of parameters and kernels. In this work, the radial basis function kernel was utilized in tuning the model to generate the requisite parameters for SVM model construction.

F. Random Forest

Random Forest, an ensemble learning algorithm renowned for its robustness and accuracy in classification tasks, comprises multiple decision trees, each allowed to grow fully without the need for pruning. The essence of Random Forest lies in aggregating the predictions of these individual trees to produce a more precise and dependable outcome, effectively mitigating concerns of overfitting commonly encountered in machine learning. Notably, the algorithm possesses the inherent capability to automatically select relevant features during training, simplifying preprocessing steps and enhancing adaptability to diverse datasets. By utilizing random subsets of features for each decision tree, Random Forest fosters diversity among trees,

enabling comprehensive exploration of the feature space.

The application of Random Forest to network anomaly detection aims to address critical challenges in the field, particularly the accurate and reliable identification of anomalous activities or potential security threats within networks [25]. The collective decision-making process of multiple trees, each contributing unique insights, empowers the algorithm to effectively differentiate between normal and malicious network behaviour. Furthermore, its ability to provide an overall estimate enhances its utility in capturing intricate patterns associated with various cyber threats. The comprehensive analysis offered by Random Forest, stemming from the amalgamation of diverse decision trees, renders it well-suited for the nuanced landscape of network anomaly detection.

G. Performance Evaluation

To evaluate the performance of the RF and MLP model, the accuracy, precision, recall, and F1-score metrics are proposed. The criterion for each of the metrics is calculated according to four main criteria which are True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) as follows:

- *True Positive (TP):* Indicates when an alarm is generated and there is an intrusion.
- *False Negative (FN):* Indicates when an alarm is not generated but there is an intrusion.
- *False Positive (FP):* Indicates when an alarm is generated but there is no intrusion.
- *True Negative (TN):* Indicates when an alarm is not generated and there is no intrusion.

Precision: Precision is a metric that focuses on the positive class and assesses how many of the predicted positives are true. It answers the question: "Of all the instances predicted as positive, how many were positive?". Equation (1) depicts the mathematical expression for the precision metrics.

$$precision = \frac{TP}{TP + FP} \tag{1}$$

Recall: Recall, also known as sensitivity or true positive rate, evaluates how many of the actual intrusion records were correctly predicted. It answers the question: "Of all the actual positives, how many were correctly predicted as positive?" The formula is shown in equation (2).

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

F-Measure (or F-Score): The F1 score is the harmonic mean of precision and recall. It provides a balanced evaluation of a classification model, taking into account both false positives and false negatives. It is particularly useful when dealing with imbalanced datasets. Equation (3) shows the mathematical formula for the F-score.

$$F - Measure = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{3}$$

Accuracy: Accuracy is a commonly used metric to evaluate the overall performance of a classification model. It measures the proportion of correctly classified instances (both true positives and true negatives) out of the total number of instances.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

IV. Result and Discussion

A. Introduction

This chapter delineates the outcomes derived from a practical investigation and implementation focused on predicting network anomaly detection from the CIC-DDoS2019 dataset source from the Mendeley machine learning repositories. The research methodology encompasses the utilization of deep learning algorithms, namely the SVM and RF algorithm. Subsequently, the constructed models undergo comprehensive validation, assessing performance metrics such as precision, recall, and F1-score.

B. Parameter Settings

The parameters outlined in Table I, play a crucial role in configuring the RF and SVM models for network anomaly detection. In the context of machine learning algorithms like RF and SVM,

parameter settings play a critical role in determining the performance and behavior of the models. For the RF, the specified parameters include the choice of the kernel and the value of the random state, whereas for SVM, parameters such as the kernel type (in this case, Radial Basis Function or RBF), the regularization parameter C, and the random state are defined. Starting with the Random Forest algorithm, the parameter setting for the random state is set to 42. This parameter controls the randomness of the algorithm, ensuring reproducibility of results across different runs. By setting a specific random state, the algorithm's random number generator will produce the same sequence of pseudo-random numbers, leading to consistent results each time the model is trained. This is particularly important for debugging, testing, and comparing different models. Moving on to the SVM algorithm, the chosen kernel is the Radial Basis Function (RBF). The RBF kernel is a popular choice in SVM models due to its ability to capture complex relationships in the data, especially in cases where the decision boundary is non-linear. By using the RBF kernel, the SVM model can effectively handle datasets with non-linear decision boundaries, making it suitable for a wide range of classification tasks. Additionally, the regularization parameter C is set to 100. The regularization parameter C in SVM controls the trade-off between maximizing the margin and minimizing the classification error. A higher value of C indicates a smaller margin and a higher penalty for misclassification, leading to a more complex decision boundary that closely fits the training data. By setting C to 100, the SVM model prioritizes achieving high accuracy on the training data while still maintaining generalization capability on unseen data.

Table I: RF and SVM Model Parameter Setting

Parameter	Value
Kernel	RBF
C	100
Random State	42

Overall, these parameter settings reflect a combination that aims to strike a balance between effective learning, model convergence, and handling the characteristics of the data used in the network anomalies detection model. Fine-tuning and experimentation with these parameters may be necessary based on the specific characteristics of the dataset and the desired performance of the model in case of further research.In the development of a network anomaly detection model, a crucial step involves incorporating a dataset for training purposes. Subsequently, the identification of anomalous incidents within the dataset was carried out utilizing RF and SVM models. To facilitate this experimental process, the Pandas library was utilized for efficient dataset manipulation. Pandas, a widely-used Python library sourced externally, offers modules designed to streamline the ingestion of datasets in various file formats. In this particular study, the dataset was formatted as a CSV (Comma Separated Values) file. Notably, the 'read_csv' function within the Pandas library proved to be highly effective in assimilating and presenting dataset attributes in a tabular structure. This functionality significantly enhances the analytical framework of the investigation, enabling a more streamlined and organized approach to data analysis. Whereas, Fig. 2 displays the result of using the pandas read_csv function.

```
In [26]: df.head()
Out[26]:
```

ned: 0	Protocol	Flow Duration	Total Fwd Packets	Total Backward Packets	Fwd Packets Length Total	Bwd Packets Length Total	Fwd Packet Length Max	Fwd Packet Length Min	Fwd Packet Length Mean	...	Active Mean	Active Std	Active Max	Active Min	Idle Mean	Idle Std	Idle Max	Idle Min	Label	Class
0	17	216631	6	0	2088.0	0.0	393.0	321.0	348.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	UDP	Attack
1	17	2	2	0	802.0	0.0	401.0	401.0	401.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	UDP	Attack
2	17	48	2	0	766.0	0.0	383.0	383.0	383.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	UDP	Attack
3	17	107319	4	0	1398.0	0.0	369.0	330.0	349.5	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	UDP	Attack
4	17	107271	4	0	1438.0	0.0	389.0	330.0	359.5	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	UDP	Attack

80 columns

Fig. 2: Depict Records

C. Data Exploration

This module, serving as a section for data exploration, facilitates the visualization of select features within the dataset. The utilization of visualizations empowers the scientific inquiry to discern inherent patterns and interrelationships among distinct variables encapsulated within the dataset.

D. Dataset Descriptive Statistics

Conducting thorough analytical research is essential for obtaining a deep statistical understanding of data characteristics, which in turn enables the precise design and execution of optimal experiments. The tabular representation in Fig. 3 provides a clear depiction of data intervals at the 25th, 50th, and 75th percentiles, obtained using Panda's data frames. This tabulated format encompasses vital statistical metrics, including count, mean, standard deviation, minimum, and maximum values. The count column enumerates instances of network anomalies records, totaling 431,371. 'Min' denotes the minimum floating value, 'std' represents the standard deviation specific to each labelled column, and 'max' indicates the maximum values in the corresponding column. Each column is associated with its unique values for descriptive statistics, offering a comprehensive analytical portrayal.

```
: df.describe()
```

	Protocol	Flow Duration	Total Fwd Packets	Total Backward Packets	Fwd Packets Length Total	Bwd Packets Length Total	Fwd Packet Length Max	Fwd Packet Length Min	Fwd Packet Length Mean	Fwd Packet Length Std
count	431371.000000	4.313710e+05	431371.000000	431371.000000	4.313710e+05	4.313710e+05	431371.000000	431371.000000	431371.000000	431371.000000
mean	13.948694	8.404856e+06	24.139117	2.472021	9.416956e+03	1.632896e+03	357.483674	294.721646	324.915327	20.208259
std	4.966712	2.126596e+07	195.888896	56.370208	3.445253e+04	1.064056e+05	320.025929	273.298705	268.577313	70.946085
min	0.000000	1.000000e+00	1.000000	0.000000	0.000000e+00	0.000000e+00	0.000000	0.000000	0.000000	0.000000
25%	6.000000	7.870000e+02	4.000000	0.000000	7.800000e+01	0.000000e+00	37.000000	6.000000	32.000000	0.000000
50%	17.000000	4.480400e+04	4.000000	0.000000	2.064000e+03	0.000000e+00	440.000000	330.000000	428.000000	0.000000
75%	17.000000	3.002508e+06	16.000000	2.000000	5.160000e+03	0.000000e+00	516.000000	516.000000	516.000000	0.000000
max	17.000000	1.199987e+08	86666.000000	31700.000000	1.526642e+07	5.842950e+07	32120.000000	2131.000000	3015.290500	2221.556200

8 rows × 77 columns

Fig. 3: Descriptive Statistics

E. Data Scaling

Normalization, commonly known as data scaling, is a crucial step in data preprocessing that significantly impacts the performance of machine learning models. Its primary goal is to standardize the scale of all features or variables present in the network anomaly dataset. This process involves adjusting the numerical values associated with these features to adhere to a standardized floating-point distribution. By doing so, normalization ensures that all features contribute equally to the model's learning process, preventing the dominance of certain features due to their larger scales. The standardized distribution achieved through normalization enhances the predictive accuracy of the model, especially in the context of early network anomaly detection. By bringing all features to a comparable scale, the model can more effectively discern patterns and anomalies within the data, leading to improved detection capabilities. In this study, the implementation of feature scaling was strategically executed using the Standard Scaler module available within the Scikit-learn machine learning framework. This module offers optimized capabilities for scaling features, ensuring efficient and effective normalization of the dataset. The utilization of the Standard Scaler module illustrates its integral role in the preprocessing pipeline of the network anomaly detection system.

F. Feature Selection

As mentioned earlier, the study employed the LDA algorithm for feature selection on the CIC-

DDoS2019 network anomaly dataset. The selected features resulting from this process are presented in Fig. 4, which is a sample code output illustrating these selected features. The threshold for the selected features was set to 20 features.

```
[n [19]: top_20_features

)ut[19]: Index(['ACK Flag Count', 'CWE Flag Count', 'URG Flag Count', 'Down/Up Ratio',
                'SYN Flag Count', 'Label', 'Protocol', 'Fwd PSH Flags',
                'RST Flag Count', 'Packet Length Mean', 'Bwd Packet Length Min',
                'Avg Packet Size', 'Fwd Packet Length Std', 'Packet Length Std',
                'Bwd Packet Length Mean', 'Avg Bwd Segment Size', 'Subflow Bwd Packets',
                'Total Backward Packets', 'Avg Fwd Segment Size',
                'Fwd Packet Length Mean'],
               dtype='object')
```

Fig. 4: LDA Feature Selection Result

G. Data Balancing

To address the class imbalance within the CIC-DDoS2019 dataset, the SMOTE data augmentation technique was employed. The reason for the application of the SMOTE data balancing was because the class labels for the attack and benign are not balanced. Hence, the SMOTE approach aimed to create a more equitable distribution between attack and benign labels. As a result of applying the SMOTE algorithm, both attack and benign labels were balanced, with each totaling 333,540 instances. The augmentation process led to the generation of additional records.

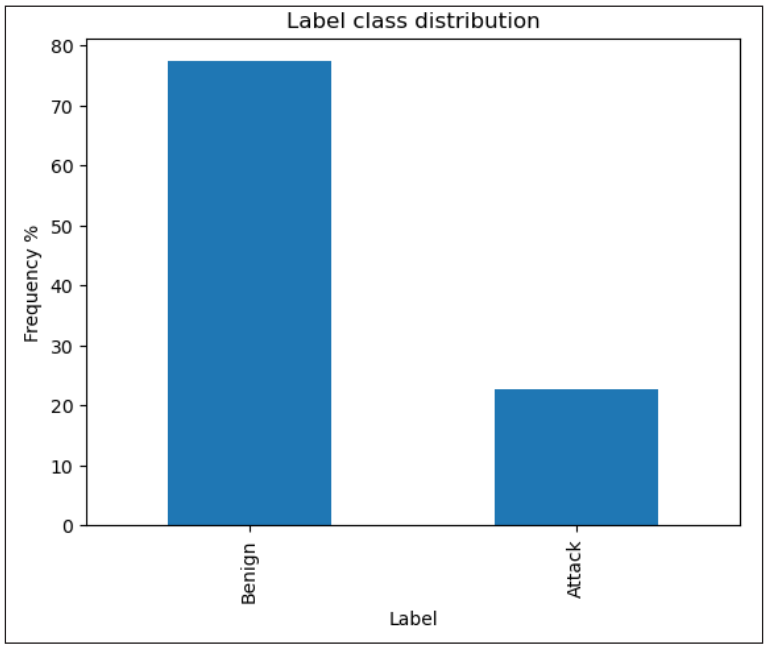

Fig. 5: Smote Data Balancing

H. Percentage Split Technique

To train the SVM and RF models, the dataset was divided into training and testing sets. The training set consisted of 70% of the total 431,371 network anomaly records, which were categorized into two classes. This training set was utilized to train the models effectively. The remaining 30% of the dataset was set aside for testing and validating the performance of the trained models.

I. Models Result

The performance of Support Vector Machine (SVM) and Random Forest (RF) models for network anomaly detection using the CIC-DDoS2019 dataset is presented in Table II. The Random Forest model achieved a training accuracy of 99.99% and a testing accuracy of 99.99%. This indicates that the RF model was able to effectively learn from the training data and generalize well to unseen data, achieving a high level of accuracy in detecting network anomalies. On the other hand, the SVM model achieved slightly lower performance with a training accuracy of 99.76% and a testing accuracy of 99.69%. While the SVM model also demonstrated strong performance, it fell slightly short compared to the RF model in terms of accuracy.

Table II: Network Anomalies Models Result (SVM and RF)

Models	Training Accuracy (%)	Testing Accuracy (%)
RF	1.0	99.99
SVM	99.76	99.69

Overall, both models performed well in detecting network anomalies, with the RF model exhibiting slightly better performance in terms of accuracy

J. Result Evaluation (Classification Report)

The SVM classification report for network anomaly detection using the CIC-DDoS2019 dataset indicates high performance across various evaluation metrics as shown in Fig. 6. With a precision of 0.9966 for class 0 and 0.9973 for class 1, the SVM demonstrates a strong ability to accurately classify instances of both normal and anomalous network traffic. Similarly, the recall scores of 0.9973 for class 0 and 0.9965 for class 1 indicate the model's effectiveness in identifying the majority of instances belonging to each class. The F1-scores, which consider both precision and recall, are also high for both classes, with values of 0.9970 for class 0 and 0.9969 for class 1. Moreover, the overall accuracy of the SVM model is reported as 0.9969, indicating its ability to correctly classify a large proportion of instances in the dataset.

```
 SVM Training Score: 0.9976788488569949
 SVM Test Score: 0.9969359571897117
SVM Classification Report:
              precision    recall  f1-score   support

           0     0.9966    0.9973    0.9970     23321
           1     0.9973    0.9965    0.9969     23023

    accuracy                         0.9969     46344
   macro avg     0.9969    0.9969    0.9969     46344
weighted avg     0.9969    0.9969    0.9969     46344

SVM AUC Score: 0.9969333329134069
```

Fig. 6: SVM Classification Report

The Random Forest classification results for network anomaly detection using the CIC-DDoS2019 dataset indicate exceptionally high-performance metrics as shown in Fig. 7. The precision, recall, and F1-score for both classes (0 and 1) are all very close to 1, indicating near-perfect classification accuracy. Specifically, for class 0, the precision, recall, and F1-score are all above 0.9996, while for class 1, they are all above 0.9996 as well. The overall accuracy of the model is reported to be 0.9998, which further emphasizes its effectiveness in accurately classifying network anomalies. The macro average and weighted average scores for precision, recall, and F1-score are also exceptionally high, all close to 0.9998.

```
 Random Forest Training Score: 1.0
 Random Forest Test Score: 0.9997842037116962
Random Forest Classification Report:
              precision    recall  f1-score   support

           0     0.9996    1.0000    0.9998      2256
           1     1.0000    0.9996    0.9998      2378

    accuracy                         0.9998      4634
   macro avg     0.9998    0.9998    0.9998      4634
weighted avg     0.9998    0.9998    0.9998      4634

Random Forest AUC Score: 0.9997897392767031
```

Fig. 7: RF Classification Report

In summary, these results suggest that both the SVM and RF models perform exceptionally well in detecting network anomalies in the CIC-DDoS2019 dataset, demonstrating high precision, recall, and accuracy across both normal and anomalous classes.

K. Result Evaluation (Receiver Operating Curve)

To comprehensively evaluate the effectiveness of the developed network anomaly detection models, the study employed the Receiver Operating Characteristic Area under the Curve (ROC-AUC) metric. ROC-AUC is a widely utilized metric for assessing classification model performance, especially in binary and multiclass scenarios. It gauges a model's ability to distinguish between positive and negative instances under varying threshold settings. The ROC-AUC curves depicted in Fig. 8 and Fig. 9 correspond to the SVM and RF models, and thus showcase the false positive rate (FPR) on the x-axis and the true positive rate (TPR) on the y-axis. TPR is synonymous with sensitivity or recall. The graphical representation elucidates the balance between TPR and FPR as the classification threshold undergoes variation. An optimal ROC-AUC curve resides in the top-left corner, indicating high sensitivity and low false positive rate, resulting in a larger area under the curve. Remarkably, the ROC-AUC metric for the SVM, and RF consistently achieved 99.99%, denoting impeccable discrimination between positive and negative instances. This signifies the models' robust performance in effectively identifying instances of network anomalies while minimizing false positives.

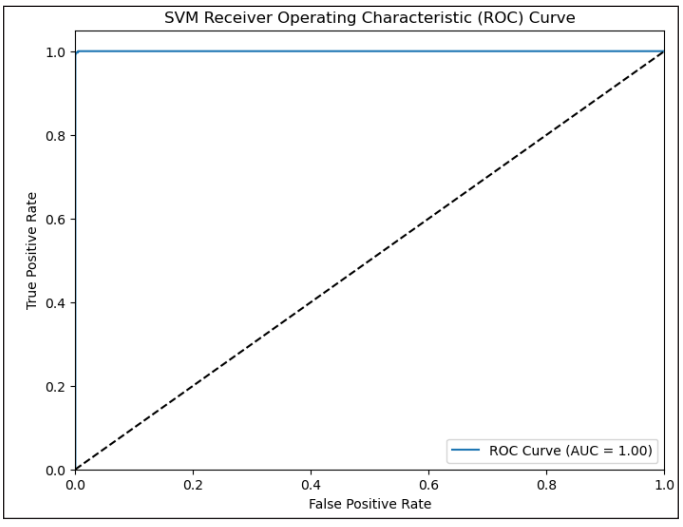

Fig. 8: SVM ROC-AUC Curve Graph

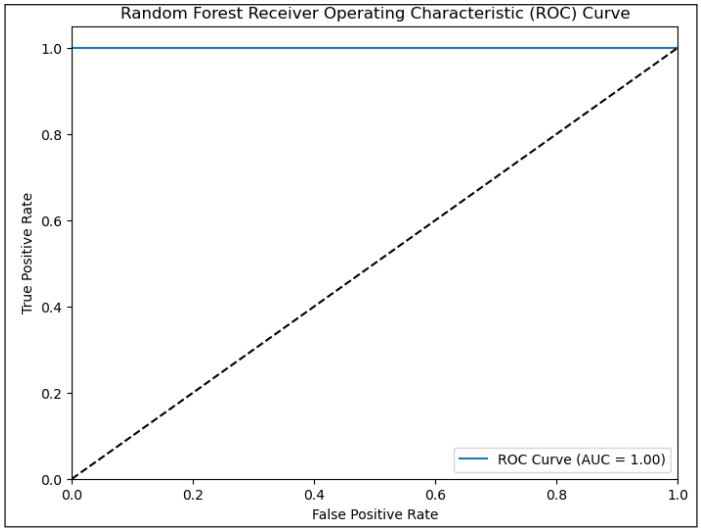

Fig. 9: RF ROC-AUC Curve Graph

In today's rapidly evolving digital environment, the proliferation of cyber threats poses a formidable challenge to the security of networked systems, a concern underscored by findings from our research survey. The urgent need to reinforce these systems against unauthorized access, data breaches, and other malicious activities emphasizes the critical importance of network anomaly detection. These models play a pivotal role in proactively identifying and mitigating potential threats by analyzing network traffic patterns, anomalous behaviours, and known attack signatures. Their deployment is indispensable for safeguarding sensitive information, preserving data integrity, and bolstering the overall resilience of interconnected infrastructures. As cyber threats continue to evolve, the ongoing development and refinement of network anomaly detection models emerge as imperative measures to outpace adversaries and uphold the security and reliability of digital networks.

To address these challenges, our study devised a systematic three-phase methodology for the development of network anomaly detection models. The initial phase involved meticulous data preparation, encompassing tasks such as the removal of unwanted characters, handling missing values, and feature selection using the LDA algorithm. Subsequently, the cleaned and

scaled dataset was inputted into machine learning algorithms, specifically SVM and RF, maintaining a 70:30 training-to-test ratio. The models underwent evaluation using precision, recall, accuracy, and F1-score metrics in the third phase of the methodology. Implementation of this research was facilitated through the utilization of the Python programming language, leveraging key third-party libraries including NumPy, Matplotlib, Pandas, TensorFlow, and Sklearn. This sequential methodology aimed to offer a structured and efficient framework for the development and evaluation of network anomaly detection models.

V. Conclusion

The study effectively developed and assessed the SVM and RF models for detecting network anomalies using the CIC-DDoS2019 dataset. The technique included feature selection by LDA, data balancing with the SMOTE algorithm, and dividing the dataset into training and testing subsets. Both models demonstrated strong performance in identifying network anomalies, with the RF model marginally surpassing the SVM model in accuracy. The RF model attained a training accuracy of 99.99% and a testing accuracy of 99.99%, whereas the SVM model reached a training accuracy of 99.76% and a testing accuracy of 99.69%. The classification results showed high precision, recall, and F1-score metrics for both models, suggesting their accurate ability to categorise regular and aberrant network traffic events. The ROC-AUC analysis verified the models' strong performance, continuously attaining 99.99% discrimination between positive and negative occurrences with very few false positives. Conclusively, the study results indicate that both SVM and RF models are successful in identifying network anomalies in the CIC-DDoS2019 dataset, demonstrating their practical utility in network security. Additional studies should investigate more feature engineering methods and model optimisations to improve efficiency and scalability in larger datasets and intricate network contexts.

A. Recommendation

This module provides prospective highlights for future application domains for the network anomaly detection model and thus identifies potential areas for advancing future research to enhance the efficacy of these models in detecting network anomalies. This exploration aims to provide insights into the broader applicability of the developed models and to suggest directions for continuous improvement in the field of network anomaly detection.

B. Application Areas

The developed network anomaly detection system exhibits significant potential for effective deployment in several key areas within the realm of cybersecurity. These areas include:

- *Enterprise Networks:* Implementing the system within enterprise networks can enhance overall cybersecurity posture by providing proactive detection and mitigation of potential network invasions. It contributes to safeguarding sensitive corporate data and maintaining the integrity of network infrastructure.
- *Critical Infrastructure Protection:* Deploying the network anomalies detection system in critical infrastructure sectors such as energy, transportation, and healthcare can bolster the security of essential services. It aids in preventing malicious activities that may pose threats to the reliable operation of critical systems.
- *Cloud Security:* Integrating the system into cloud environments will ensure the continuous monitoring and protection of cloud-based applications and services. It adds an extra layer of defence against unauthorized access and data breaches in cloud computing architectures.

C. Suggestions for Further Research

After an extensive experimental study and review of several kinds of literature relating to the developed

network anomalies detection models, this study suggests future research as follows:

- *Ensemble Approaches:* Future studies can investigate the potential of ensemble methods by combining the strengths of different models, such as KNN, Multilayer Perceptron, etc. Ensemble techniques, like stacking or bagging, could potentially improve overall performance by leveraging the complementary strengths of diverse models.
- *Dynamic Adaptability:* Future studies can also explore the development of network anomaly detection systems that dynamically adapt to evolving network environments. This could involve the integration of reinforcement learning or other adaptive techniques to enhance the system's ability to recognize novel network anomalies or intrusion patterns.

References

[1] M. Saharkhizan, A. Azmoodeh, A. Dehghantanha, K. K. R. Choo, and R. M. Parizi, "An ensemble of deep recurrent neural networks for detecting IoT cyber-attacks using network traffic," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 8852-8859, 2020.

[2] A. B. Nassif, M. A. Talib, Q. Nasir, and F. M. Dakalbab, "Machine learning for anomaly detection: A systematic review," *IEEE Access*, vol. 9, pp. 78658-78700, 2021.

[3] O. N. Nyasore, P. Zavarsky, B. Swar, R. Naiyeju, and S. Dabra, "Deep packet inspection in industrial automation control system to mitigate attacks exploiting Modbus/TCP vulnerabilities," in *2020 IEEE 6th Intl Conference on Big Data Security on Cloud (BigDataSecurity),* IEEE, 2020, pp. 241-245.

[4] N. Elmrabit, F. Zhou, F. Li, and H. Zhou, "Evaluation of machine learning algorithms for anomaly detection," in *2020 International Conference on Cyber Security and Protection of Digital Services (Cyber Security),* IEEE, Jun. 2020, pp. 1-8.

[5] A. Bommert, X. Sun, B. Bischl, J. Rahnenführer, and M. Lang, "Benchmark for filter methods for feature selection in high-dimensional classification data," *Computational Statistics & Data Analysis*, vol. 143, p. 106839, 2020.

[6] S. Wang, J. F. Balarezo, S. Kandeepan, A. Al-Hourani, K. G. Chavez, and B. Rubinstein, "Machine learning in network anomaly detection: A survey," *IEEE Access*, vol. 9, pp. 152379-152396, 2021.

[7] M. Ahmed, A. N. Mahmood, and J. Hu, "A survey of network anomaly detection techniques," *J. Netw. Comput. Appl.,* vol. 60, pp. 19-31, 2016.

[8] M. Usama *et al.*, "Unsupervised machine learning for networking: Techniques, applications and research challenges," 2017, arXiv preprint, arXiv:1709.06599.

[9] B. Lindemann, B. Maschler, N. Sahlab, and M. Weyrich, "A survey on anomaly detection for technical systems using LSTM networks," *Computers in Industry*, vol. 131, p. 103498, 2021.

[10] B. Venkatesh, and J. Anuradha, "A review of feature selection and its methods," *Cybernetics and Information Technologies*, vol. 19, no. 1, pp. 3-26, 2019.

[11] V. Bolón-Canedo, E. Ataer-Cansizoglu, D. Erdogmus, J. Kalpathy-Cramer, O. Fontenla-Romero, A. Alonso-Betanzos, and M. Chiang, "Dealing with inter-expert variability in retinopathy of prematurity: A machine learning approach," *Comput. Methods Progr. Biomed.*, vol. 122, no. 1, pp. 1-15, 2015.

[12] M. Nicolau, and J. McDermott, "Learning neural representations for network anomaly detection," *IEEE Transactions on Cybernetics*, vol. 49, no. 8, pp. 3074-3087, 2018.

[13] D. Kwon, K. Natarajan, S. C. Suh, H. Kim, and J. Kim, "An empirical study on network anomaly detection using convolutional neural networks," in *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS),* IEEE, 2018, pp. 1595-1598.

[14] B. J. Radford, L. M. Apolonio, A. J. Trias, and J. A. Simpson, "Network traffic anomaly detection using recurrent neural networks," 2018, arXiv preprint, arXiv:1803.10769.

[15] G. Fernandes, J. J. Rodrigues, L. F. Carvalho, J. F. Al-Muhtadi, and M. L. Proença, "A comprehensive survey on network anomaly detection," *Telecommunication Systems*, vol. 70, pp. 447-489, 2019.

[16] A. Nagaraja, U. Boregowda, K. Khatatneh, R. Vangipuram, R. Nuvvusetty, and V. S. Kiran, "Similarity based feature transformation for network anomaly detection," *IEEE Access*, vol. 8, pp. 39184-39196, 2020.

[17] S. Naseer, Y. Saleem, S. Khalid, M. K Bashir, J. Han, M. M. Iqbal, and K. Han, "Enhanced network anomaly detection based on deep neural networks," *IEEE Access*, vol. 6, pp. 48231-48246, 2018.

[18] A. H. Hamamoto, L. F. Carvalho, L. D. H. Sampaio, T. Abrão, and M. L. Proença Jr, "Network anomaly detection system using genetic algorithm and fuzzy logic," *Expert Systems with Applications*, vol. 92, pp. 390-402, 2018.

[19] D. Kwon, H. Kim, S. C. Suh, I. Kim, and K. J. Kim, "A survey of deep learning-based network anomaly detection," *Cluster Computing*, vol. 22, pp. 949-961, 2019.

[20] M. Said Elsayed, N. A. Le-Khac, S. Dev, and A. D. Jurcut, "Network anomaly detection using LSTM based autoencoder," in *Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks*, 2020, pp. 37-45.

[21] W. Xu, J. Jang-Jaccard, A. Singh, Y. Wei, and F. Sabrina, "Improving performance of autoencoder-based network anomaly detection on NSL-KDD dataset," *IEEE Access*, vol. 9, pp. 140136-140146, 2021.

[22] D. K Choubey, M. Kumar, V. Shukla, S. Tripathi, and V. K. Dhandhania, "Comparative analysis of classification methods with PCA and LDA for diabetes," *Current Diabetes Reviews*, vol. 16, no. 8, pp. 833-850, 2020.

[23] L. Yang, X. Liu, F. Nie, and Y. Liu, "Robust and efficient linear discriminant analysis with $L_{2,1}$-norm for feature selection," *IEEE Access*, vol. 8, pp. 44100-44110, 2020.

[24] D. Wang, and G. Xu, "Research on the detection of network intrusion prevention with SVM-based optimization algorithm," *Informatica*, vol. 44, no. 2, 2020.

[25] Z. Liu, N. Su, Y. Qin, J. Lu, and X. Li, "A deep random forest model on spark for network intrusion detection," *Mobile Information Systems*, pp. 1-16, 2020.