Anomaly-Based Intrusion Detection Systems Using Machine Learning

Alsamir Alqahtani[1,*], Hanan AlShaher[1]

[1]Department of Computer Science, College of Computer and Information Science, Majmaah University, Al-Majmaah 11952, Saudi Arabia

Emails: 441104466@s.mu.edu.sa; h.alshaher@mu.edu.sa

Abstract

With the increased use of the Internet, unauthorized access has increased, allowing malicious users to hack networks and carry out malicious activities. One of the essential modern approaches in today's cybersecurity efforts is the limitation of access by suspect users. In this study, the approach toward real-time intrusion detection was to consider behavioral patterns of past users on the network. We classified the users as two categories: intervention and non-intervention, and employed the machine learning techniques Artificial Neural Networks [ANN], Support Vector Machines [SVM], and Decision Trees [DT]. The Decision Trees model was chosen as it had a mature capability concerning complex pattern recognition and an enhancement capability of the intrusion detection systems. The efficiency of these algorithms is examined via the key performance metrics: confusion matrix, F1-score, and Area under the Curve [AUC]. Decision Tree, which came up as the best model for both the training and testing phases, produced an outstanding F1-score of 99.96% and AUC score of 99.93% in the testing phase. SVM and ANN gave good results; the F1 scores of SVM and ANN in the testing phase were 92.76% and 93.33%, while the AUC was 90.57% and 94.78%, respectively. This research will enlighten us on the influence of machine learning on the scope of intrusion detection, fostering more development efforts toward more responsive and dynamic intrusion detection systems. The comparative evaluation of these models will help in providing vital information for the further enhancement of cybersecurity strategies, ensuring better defenses against these ever-evolving cyber threats.

Keywords: Machine Learning; Decision Tree; Support Vector Machine; Artificial Neural Networks

1. Introduction

Improving the usage of the Internet opened new hazards that presently promise to be gravely threatening too many forms of businesses and organizations. The penetration of hackers and cybercriminals by all possible cyber mechanisms finally causes financial and infrastructural losses to corporations, institutions, and people. Such events are not only costly but also highly damaging to very large infrastructures of network systems [1].

So far, these traditional defenses against cyber problems—firewalls, data encryption, user authentication, and security software—have not worked. In fact, in practice, these measures do not really shield an individual from a variety of digital threats because the sophistication of the attack techniques is more sophisticated than the development of those traditional security systems [1-2].

DOI: https://doi.org/10.54216/JCIM.140102
Received: January 06, 2024 Revised: Marach 18, 2024 Accepted: May 19, 2024

For instance, even though a firewall controls the access between two networks, it cannot indicate a security breach from within, a reason why the urgency to employ systems such as machine learning–based intrusion detection systems (IDS) cannot be emphasized more to help reinforce the security protocols at play [3].

As a result, cybersecurity has emanated as a serious challenge in the field of technology and high within the Internet ecosystem. Relevant and effective strategies to counter these digital aggression agents are endlessly being sought by institutions, while at the same time, their cyber adversaries never grow stale in their innovation as new vulnerabilities continue to become manifest, thereby endangering the fabric of the institutions and the society at large [4]. This is usually a dual-pronged strategy entailing preprocessing and filtering of network traffic to identify and mitigate known threats and patterns and deep analyses of residual data to detect novel and emerging threat patterns, thereby enhancing the security and integrity of network systems.

Typically, detection of cyber-security threats within a network is done using any of two methodologies: (1) preprocessing and filtration of network traffic according to known risk profiles and known attack vectors, and (2) careful scrutiny of the remaining data in search of new and unknown patterns. The current work will emphasize the second approach, which is supported by the fact that records of user activity and network access and the actions taken subsequently are available in a detailed manner through the logs of the network. Nevertheless, the analysis of user behavior is time-consuming, very hard, and quite erroneous since many biases and human errors penetrate this manual analysis.

In view of these challenges, artificial intelligence—with a more specific focus, machine learning—provides a realistic and efficient solution. The concept of machine learning allows for the autonomous handling and analysis of voluminous data in the identification of patterns and then quick processing by machines. Ultimately, ML could categorize users based on behavioral patterns, raising an alarm for network admins on potential malicious actors early enough for preemptive detection, and eventually being restricted from gaining network access—thus, effectively minimizing the threat surface [5].

Machine learning algorithms are highly complex computational entities that learn adaptively from data. It is considered that they have the purpose of emulating the human cognitive process. They have become in this way basic tools in this era of big data in most sectors, including finance, media, and computational biology in addition to the biomedical and health disciplines, which show versatility and efficacy of ML methodologies [6].

An intrusion detection system is one of the most effective ways in cybersecurity because it can detect malicious users even before they enter the system. Such malicious users' activities cannot be located in the traditional way, which relies on human monitoring, until they damage the network. The normal way also depends mostly on the behavior of humans, the fault of many errors and biases. In contrast, the use of ML can overcome such issues [7].

An intrusion detection is categorized into four types: anomaly, signature, pattern, and henbit. This paper adopts anomaly detection, since signature and pattern detections do not cover all the cases, and the cost of henbit is very high. Hence, the best way to identify intrusion is detecting anomalies. A variety of models, which include Artificial Neural Networks (ANN), Support Vector Machines (SVM), and Decision Trees (DT), are presented. The reason for choosing these models is their wide application and they have also proved to be very effective [8]. Tests that were carried out on actual datasets proved beyond any other reasonable doubt that the algorithms are highly effective in applying them to real applications [9].

These algorithms are evaluated using measures like the F1-score and the confusion matrix. Since the F1-score is an optimal measurement of the accuracy of the model, especially when the class distributions are imbalanced, it becomes the harmonic average of precision and recall. On the other hand, the confusion matrix gives a tabular way describing the performance of an algorithm, and it helps us understand in what way the classification model can confuse the different classes. Both tools of evaluation will be very useful in analyzing, comparing, and ultimately choosing the most effective model for the identification of malicious users.

Objectives

DOI: https://doi.org/10.54216/JCIM.140102
Received: January 06, 2024 Revised: Marach 18, 2024 Accepted: May 19, 2024

The aim of this research is really to integrate a few of the machine learning strategies, chosen on the basis of measures of accuracy, to predict the user data pattern. These would encompass the selection of an algorithm that has the maximum accuracy in early warning signs of malicious users trying to access the system. This would be done through:

1- Collecting and analyzing the dataset of users who have accessed the network.

2- Implementing and assessing various classification algorithms to identify the most effective one.

3- Evaluating the accuracy of the chosen model using a test dataset.

Research questions

1. Which is the most accurate ML algorithm for detecting malignant network entry?
2. Does data pre-processing optimize the accuracy of the results?
3. How do different ML algorithms perform under varying network conditions and attack scenarios?
4. What is the impact of real-time data analysis on the detection capabilities of ML algorithms for network intrusion?

2. Literature Review

Alqahtani et al. (2020) applied a range of prevalent ML classification algorithms to the challenge of intrusion detection. Their arsenal included Bayesian Network, Naive Bayes classifier, Decision Tree, Random Decision Forest, Random Tree, Decision Table, and Artificial Neural Network. They evaluated these models based on performance metrics, such as precision, recall, f1-score, and accuracy, and conducted numerous tests on cybersecurity datasets that included various cyberattack categories. Their experiments identified the Random Forest algorithm as the most accurate at the conclusion of their testing. [10]

Alzahrani et al. (2021) tested the ML algorithms for detecting the malicious activities and analyzing the network traffic. In their research work, they proposed to develop the Decision Trees, Random Forest, and XGBoost algorithms on the dataset called NSL-KDD. After the experiment, they obtained an accuracy rate of 95.95% [11].

Rokade et al., proposed implementing the ML algorithms for detection purposes of any intruder in the network. For example, two types of major detection techniques are proposed: anomaly-based and misuse-based detection. These algorithms are tested by implementing the Support Vector Machine, Naive Bayes, and Artificial Neural Network algorithms using several datasets for the purpose of testing the performance of these systems under a real-time network [12].

Sangkatsanee et al. (2021) introduced a real-time intrusion detection method based on supervised ML. They report the results of their experiments, which have demonstrated that the Decision Tree technique provided better accuracy as compared to other models. In this respect, they have designed an RT-IDS in which, with the use of the Decision Tree method, the streams of network data are classified into either normal or attack-related and then achieved the detection accuracy of more than 98% for the time of fewer than two seconds [13].

Raghuvanshi et al. (2022) preprocessed the data of the NSL-KDD dataset at first. They developed the data by converting all the symbols of the NSL-KDD dataset into a numerical expression. Next, they performed the feature extraction by using Principal Component Analysis (PCA) over the resultant dataset and classified the same dataset using Random Forest, Linear Regression, and Support Vector Machine (SVM). The results of the developed models state that the developed model has the performance results, where the SVM has shown high accuracy of 98%, Random Forest of 85%, Linear Regression of 78% while measured with the accuracy, precision, and recall metric to show good performance [14].

Faker and Dogdou made an integration of big data analytics into a deep learning framework to make the intrusion detection system more efficient. They made an analysis using three classifiers in total for the classification of the network traffic data, which was Random Forest and Gradient Boosting Tree

DOI: https://doi.org/10.54216/JCIM.140102
Received: January 06, 2024 Revised: Marach 18, 2024 Accepted: May 19, 2024

in the form of an ensemble algorithm; the third one was the Deep Feed Forward Neural Network (DNN) classifier. The primary result of this approach was that the DNN classifier showed high precision with an accuracy of 99.16% in binary cases and 97.01% in multiclass when analyzed against the UNSW NB15 dataset. The classifier trained on the CICIDS2017 dataset indicated that, in case of binary classification, the GBT classifier is best, which is correct to 99.99%, whereas, in the case of multiclass classification, the DNN classifier is actually 99.56% accurate when it is analyzed against the dataset [15].

Shin and Kim developed intrusion detection models that aim to label data classes under the categories "attack" or "normal" to detect unusual patterns and network anomalies using ML techniques. They tested the models the authors proposed in the ADFA Linux dataset. Part of the dataset was developed through system call traces to emulate interactions with a contemporary attack operating system. The authors created three kinds of ML mechanisms: that is, SVM, KNN, and Logistic Regression, and made three numbers of models using the AUC technique. They have found the rates of accuracy for three models, such as 78% in the case of the SVM model, 82% in the case of the KNN model, and 84% in the Logistic Regression model, and the rate is higher than that of the previous similar research [16].

Wester et al. (2021) detected network intrusions through the identification of anomalous behavior by the use of the TAN algorithm, using datasets KDD and UNSW-NB15. This study resulted in the ability of the algorithm to attain 99% accuracy while keeping the false positive rate at or below 0.5%, which is really very remarkable [17].

Barbhuiya et al. (2020) in an independent study affirmed that: This model gives way to the advanced scheme, which can be developed by DroidLight, a malware detection tool, in such a manner that the proposed setting sounds an alert, if the deviation from learned normal trends is very huge to ignore. These authors developed three malware programs and utilized these to test Droidlight's last efficacy. Their proposed model was cross-validated by real users in real devices to make up an authentic operating environment. The accuracies of their proposed model were found in the range of 93% and 100% [18].

Alameidy et al. (2019) —Developed an advanced ID framework using a multi-objective GWO. The purposely designed feature selector was used in such a manner to pick out important features in a dataset, therefore making the detection model used to be accurate. The developed scenarios attacked the classification systems using an SVM classifier. It was first tested on 20% of the NSL-KDD set, where the authors reported that the results were promising. The proposed system showed high-classification accuracy of 93.64, 91.01, 57.72, and 53.7% for DoS, Probe, R2L, and U2R attacks, respectively [19].

As a product of the NSL-KDD data, Yihunie et al. developed the network intrusion detection system to compare the performance of ML algorithms. Within this work, the main classification goal was checking the performance of algorithms based on their classification accuracy for classes "normal" and "anomalous" network traffic. Such binary classifiers as Stochastic Gradient Descent (SGD), Random Forests, Logistic Regression, Support Vector Machine (SVM), and the Sequential Model were developed with the support of the Keras Library. On the other hand, every classifier was thoroughly tested to check which one could optimally perform in this application. Thus, they reported the classifier's accuracies as follows: 97.19 for SGD, 87.31 for Logistic Regression, 99.80 for Random Forest, and 97.55 for SVM; that is to say, 95.497 with the Sequential Model. In this regard, the Random Forest classifier has the best accuracy among the classifiers tested, which is quite significant. More so, through an F-score analysis with $\beta = 9$, it shows that the classifier occurs to be highly robust and capable of an accurate classification with the Random Forest algorithm. It was one of the most competent algorithms for network intrusion detection. [20]

In a related study, Bhavsar et al. developed a novel Intrusion Detection System, using a deep learning architecture termed the Pearson Correlation Coefficient and Convolutional Neural Network, or PCC-CNN. Their research checked the work of this model with three potential and much widely used datasets: NSL-KDD, CICIDS2017, and IOTID20. The first model evaluated was the training and testing of the five different ML models under the Pearson Correlation Coefficient—SVM, LR, LDA, KNN, and CART. The performance of the models was evaluated using an extensive battery of metrics that included accuracy, precision, recall, and F1 score, which helps to give an exhaustive status of the intrusion detected. The main findings showed that KNN and CART in both binary and multi-

DOI: https://doi.org/10.54216/JCIM.140102
Received: January 06, 2024 Revised: Marach 18, 2024 Accepted: May 19, 2024

classification scenarios resulted in the maximum best accuracy, which approximated at around 98% and 99%, respectively. On the other hand, the LR, LDA, and SVM models resulted in relatively low accuracies, especially when they are involving multiclass classification tasks, varying from around 78% to 88%. They further compared their proposed PCC-CNN model with other ML models, which found, across the three datasets, produced a general accuracy of 99% in binary classification tasks. Interestingly, this study showed that the PCC-CNN model can produce highly accurate predictions of intrusions in cases where the datasets have imbalanced cases of assault.[21].

Siganos et al. (2023) demonstrated an AI-powered IDS system designed for the IoT ecosystem. Features that make this model explainable include an SHapley Additive exPlanations approach, embedded with models based on both ML and DL approaches. The curtains of the black-box working methodology of the IDS were made transparent. Assessment of this IDS showed its talent not only in detection performance but also in terms of Explainable AI, which is an emerging requirement for complex AI systems. Empirical Evaluation of the Proposed IDS System: Two balanced datasets have been used for the empirical evaluation of the proposed IDS system, IEC 60870-5-104 included. Ten different ML algorithms have been used in the present work: Naive Bayes, SVM, Linear SVM, RBF, Decision Trees, RF, XGBoost, AdaBoost, Logistic Regression, Quadratic Discriminant Analysis, and DNN. Of all these, RF came out to be maximum accurate, giving an F1-score of 66% [22].

The performance results from Espes et al. indicate promise: more crucially, the false-positive rate was minimized, and the percentage of accuracy reached high levels. These algorithms were evaluated in terms of precision using F1-Score, a measure reflecting both precision and recall, and they were found to have illustratively resulted in 100% accuracy. These results hence suggest the potential of their approach in drastically bringing up the reliability and efficiency for cyberattack repercussion detection in industrial setups.

3. Methodology

The present research elaborates on a wide range of machine learning algorithms used within this dissertation. It clearly classifies and distinguishes between supervised and unsupervised learning models to clarify the respective characteristics and special applications for which each is best suited.

Classification Algorithms

Decision Tree (DT)

A decision tree is a flowchart-like structure that helps in making various predictions by charting a hierarchy of features and outcomes that will together serve the purpose. Such a model could assist in decision-making through the identification of the optimal paths to the desired outcomes and, at the same time, allow for the provision of alternatives for uncertain outcomes. The choice of such paths is done with respect to the degree to which such paths are accurate and the strength of the classification rules that they resultantly explain [23].

Thus, a decision tree is operationally based on a principle of stepwise, progressive partitioning of full-scale data into ever smaller and smaller sets until a final partition, or a final known terminal node, called a leaf, is reached. It means that it is possible to distill huge arrays of data into insights which can be managed. At this point, it is important to mention that other types of algorithms, such as ID3, C4.5, and CART, and not Sprint, are applied in order to implement different types of decision trees with different rules and various interpretations but with a good extent of transparency and simplicity in mechanisms [24].

Decision trees find extensive applications across a multitude of domains including machine learning, image processing, and pattern recognition. They sequentially link an array of elementary tests, each juxtaposing a numerical attribute against a corresponding threshold, to form an integrated model that navigates through the data to arrive at a decision.

Despite their extensive applications, decision trees are not without limitations. One major drawback is the necessity of sorting all numerical values to identify optimal splitting points for the tree. Additionally, they are sensitive to slight variations in the training data, which can result in significant changes to the structure of the tree, highlighting a lack of robustness. Moreover, the complexity of the generated tree can sometimes escalate, rendering it difficult to interpret [25].

DOI: https://doi.org/10.54216/JCIM.140102
Received: January 06, 2024 Revised: Marach 18, 2024 Accepted: May 19, 2024

An illustrative example of a decision tree is provided in Figure (1), which offers a visual representation of the model's structure and decision paths.

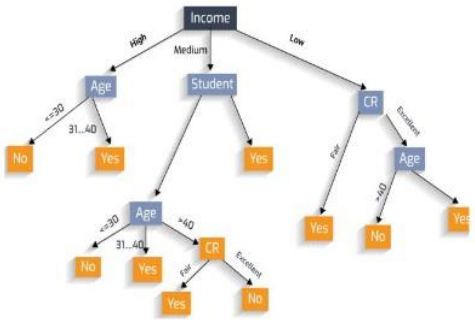

Figure 1: Decision Tree [26]

Support Vector Machine (SVM)

Support Vector Machine (SVM) is indeed a type of supervised learning, widely recognized for its application in both prediction and classification tasks. As previously indicated, supervised learning entails training models on data that is distinctly partitioned into inputs (features) and outputs (labels). SVM stands out as one of the most robust and powerful algorithms in the realm of machine learning.

The SVM algorithm treats the data as a set of points in an n-dimensional space, where n is the number of features. Most importantly, such an algorithm very much aims at identifying the best separating boundary, which is visited as the optimal hyperplane between classes of data points. It is the hyperplane that separates the datasets into their classes. In this manner, SVM aids the process of data partitioning and classification, a processing step in mining the trends and patterns from the data. It helps in effective segregation and categorization of the data into classes with similar characteristics, thus meaningful patterns drawn out from the data [27].

The working mechanism of the support vector machine is explained by the diagram below (Figure 2).

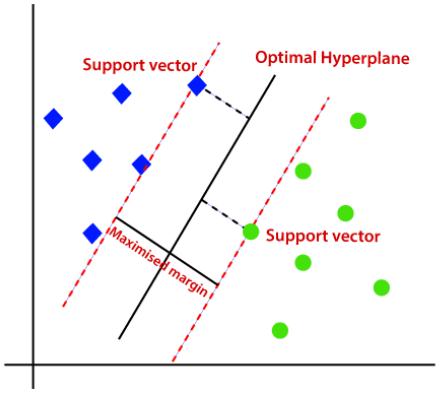

Figure 2: Support Vector Machine (SVM) [27]

Artificial Neural Network (ANN)

The ANN is a computing model that reproduces the biological nervous system's information processing. For example, just as it is in the network of neurons of the human brain, the architecture of an ANN is made up of units that are connected to work cooperatively for the solution and resolution of problems. Generally, an ANN is formed in three main layers: the input layer, which provides the initial data; one or more hidden layers, in which the data is organized and changed; and the output layer, returning the final results of the computation. One of the most intricate algorithms in machine learning science, ANNs are particularly praised in performing multidimensional processes. Moreover, these assemblies have a great history in computing science, and its early progress goes back to the 1940s. Their value was significantly proven by the development of the backpropagation algorithm, one of the main methods built in the 1980s; it allows the network's connection weights within the

DOI: https://doi.org/10.54216/JCIM.140102
Received: January 06, 2024 Revised: Marach 18, 2024 Accepted: May 19, 2024

hidden layers to be adjusted. This is required in the event that the collected output of the network were to deviate from the target or expected values. [28]

Visual representations, such as that provided in Figure (3), elucidate the operational framework of neural networks

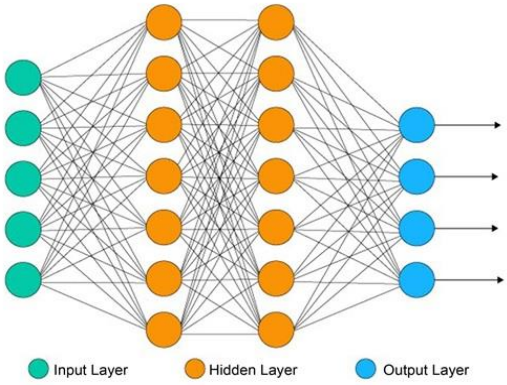

Figure 3: ANN diagram [29]

We will detect the anomaly, according to the collected data set, where every record(user) has many features, and one class which refer to the user if he is hacker or not.

We will train the model on all records and at the end the model will be able to detect the hacker according to his features.

Performance Evaluation Metrics

Model Outcome Categorization

The evaluation segment of the research assesses the efficiency of the classification model in terms of its predictive accuracy. The model assigns either a positive or negative label to each instance within the data set, categorizing each instance as follows:

- **True Positives (TP):** Actual positive samples that are predicted as positive by the model.
- **False Negatives (FN):** Actual positive samples that are predicted as negative by the model.
- **True Negatives (TN):** Actual negative samples that are predicted as negative by the model.
- **False Positives (FP):** Actual negative samples that are predicted as positive by the model.

$$\text{accuaracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

The formula of recall:

$$\text{recall} = \frac{TP}{TP + FN} \quad (2)$$

The formula of precision:

$$\text{precision} = \frac{TP}{TP + FP} \quad (3)$$

The formula of F1-score:

$$\text{F1} - \text{score} = 2.\frac{\text{precision}.\,\text{recall}}{\text{precision} + \text{recall}} \quad (4)$$

DOI: https://doi.org/10.54216/JCIM.140102
Received: January 06, 2024 Revised: Marach 18, 2024 Accepted: May 19, 2024

Evaluation Metrics

F1 Score: The models are evaluated in terms of their predictive power through the F1 score, which is a harmonic mean between precision and recall. It balances the two, being more useful for an imbalanced class distribution. The F1 scores yield through the comparison of expected and actual outcomes about a test set are important characteristics to compare the performance between the algorithms. The model in the field that makes the real predictions will be that with the highest F1 score.

The Area Under the Curve (AUC): The AUC evaluates the ability of a model to discriminate classes, which can be summarized by the receiver operating characteristic. A higher AUC indicates a better ability to distinguish between positive and negative classifications. An AUC of 1 shows a perfect classification, as all positive and negative instances are correctly identified. On the other hand, an AUC of 0 shows the classifier is invertingly classifying all the instances, settings on mislabeling positives as negatives and negatives as positives.

confusion matrix: The confusion matrix is shown to be an important tool in the process of classification, for binary and multiclass classification problems. An example of a confusion matrix is shown in Figure (4), as it allows for the presentation of performance metrics graphically for binary classification.

		Predicted	
		Negative **(N)** -	Positive **(P)** +
Actual	Negative -	True Negative **(TN)**	**False Positive (FP) Type I Error**
	Positive +	**False Negative (FN) Type II Error**	True Positive **(TP)**

Figure 4: Confusion Matrix

Research Design

The present study employs a quantitative research design, applying machine learning in classifying network users based on their activity logs. Proper classification of the action of the users is done into the two major classes: intervention and non-intervention. The study further evaluates performance metrics like the confusion matrix, F1 score, and the area under the curve to gauge the effectiveness of the classifier in the two different classes. The machine-learning algorithms applied in this study are Artificial Neural Networks, Support Vector Machines, and Decision Trees.

Data Collection

The dataset was obtained from Kaggle, named CIDDS-001-external-week and contains network user logs that embody distinctive user activity within a networked environment. More specifically, the dataset contains 172,838 instances with 17 different-columned data, and the target variable is classified into three classes: normal, suspicious, and unknown. Precisely, their distribution is 107,344 suspicious, 49,606 in the normal, and 15,888 in the unknown.

	Date first seen	Duration	Proto	Src IP Addr	Src Pt	Dst IP Addr	Dst Pt	Packets	Bytes	Flows	Flags	Tos	class	attackType	attackID	attackDescription
0	2017-03-14 17:43:57.172	81412.697	TCP	EXT_SERVER	8082	OPENSTACK_NET	56978.0	3057	2.1 M	1	.AP...	0	normal	---	---	---
1	2017-03-14 17:43:57.172	81412.697	TCP	OPENSTACK_NET	56978	EXT_SERVER	8082.0	4748	2.5 M	1	.AP...	0	normal	---	---	---
2	2017-03-14 17:43:26.135	81504.787	TCP	EXT_SERVER	8082	OPENSTACK_NET	56979.0	8639	9.1 M	1	.AP...	0	normal	---	---	---
3	2017-03-14 17:43:26.135	81504.787	TCP	OPENSTACK_NET	56979	EXT_SERVER	8082.0	12024	10.3 M	1	.AP...	0	normal	---	---	---
4	2017-03-14 18:17:09.005	82100.692	TCP	EXT_SERVER	8082	OPENSTACK_NET	51649.0	11012	27.2 M	1	.AP.S.	0	normal	---	---	---
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
172833	2017-03-16 11:42:01.296	518988.907	TCP	OPENSTACK_NET	49939	EXT_SERVER	8082.0	17446	10.6 M	1	.APRS.	0	normal	---	---	---
172834	2017-03-16 12:10:17.340	517292.865	TCP	EXT_SERVER	8082	OPENSTACK_NET	58749.0	9406	3.4 M	1	.AP.S.	0	normal	---	---	---
172835	2017-03-16 12:10:17.340	517292.865	TCP	OPENSTACK_NET	58749	EXT_SERVER	8082.0	15510	10.3 M	1	.APRS.	0	normal	---	---	---
172836	2017-03-16 15:52:58.342	503931.863	TCP	EXT_SERVER	8082	OPENSTACK_NET	62605.0	13362	5.2 M	1	.AP.S.	0	normal	---	---	---
172837	2017-03-16 15:52:58.342	503931.863	TCP	OPENSTACK_NET	62605	EXT_SERVER	8082.0	20262	17.8 M	1	.APRS.	0	normal	---	---	---

172838 rows × 16 columns

DOI: https://doi.org/10.54216/JCIM.140102
Received: January 06, 2024 Revised: Marach 18, 2024 Accepted: May 19, 2024

Figure 5: Sample of CIDDS Dataset

Data Preprocessing

The preprocessing steps undertaken to prepare the dataset for analysis included several critical actions:

- **Column Removal:** Columns such as 'attackType', 'attackID', and 'attackDescription' were removed due to lack of data. The 'Flows' and 'TOS' columns, which held constant values across all entries, were also dropped. The 'Date first seen' column was eliminated as it was deemed irrelevant to the analysis objectives.
- **Label Encoding:** To facilitate machine learning analysis, label encoding was applied to the 'Dst IP Addr', 'Src IP Addr', 'Proto', and target class columns to convert categorical data into a machine-readable format.
- **Data Splitting:** The final dataset was divided into training and testing sets, with 80% of the data allocated for training and 20% for testing, ensuring a substantial amount of data for model training while still providing an adequate test set for model evaluation.

These preprocessing steps were designed to optimize the dataset for the machine learning tasks, focusing on improving model accuracy and computational efficiency.

Feature Selection

In this research, the feature selection process was integral to refining the predictive model by isolating the most influential variables from the dataset. Initially, the dataset comprised several features; however, for the purpose of this study, the following features were considered based on their relevance to the classification task: "Proto", "Src IP Addr", "Src Pt", "Dst IP Addr", "Dst Pt", and "Packets".

We chose features that have a direct impact on the network traffic data, since they will possibly affect the user class in network intrusion detection; the target variable, 'class', defined for each instance, will classify the target variables into three classes: normal, suspicious, and unknown, for the supervised learning model to carry out a task.

Rationale for Selection:

- **Protocol ("Proto")**: Type of protocol used in data packets; it can be critical in identifying patterns that are specific to normal and malicious activities.
- **Source IP Address ("Src IP Addr") and Destination IP Address ("Dst IP Addr")**: The source and destination IP addresses provide insight into the origin and destination of the traffic, which is crucial for anomalies in network behaviors being detected.
- **Source Port ("Src Pt") and Destination Port ("Dst Pt")**: Understanding specific services or applications that were being targeted or originated by the attack; this is a tenet in pattern recognition in intrusion detection.
- **Packets ("Packets")**: The number of packets sent in the connection, which can be indicative of the volume and intensity of the communication across the network.

The selected features will then be used to train the machine learning classifiers to classify the network users effectively. This was based on the fact that these were going to be able to carry enough information capable of distinguishing between different kinds of network behaviors, focusing more on malicious activities.

4. Results

Performance of three models—Decision Tree, Support Vector Machine (SVM), and Artificial Neural Network (ANN)—was thoroughly analyzed. Metrics of performance were made on these classifiers, which classified the network traffic data into three classes of suspicious, unknown, and normal.

Training Phase Results:

DOI: https://doi.org/10.54216/JCIM.140102
Received: January 06, 2024 Revised: Marach 18, 2024 Accepted: May 19, 2024

Table 1: Training Phase Results

Model	**AUC (Training) %**	**F1 (Training) %**
Decision Tree	99.99%	99.99%
SVM	90.69%	92.79%
ANN	94.78%	93.34%

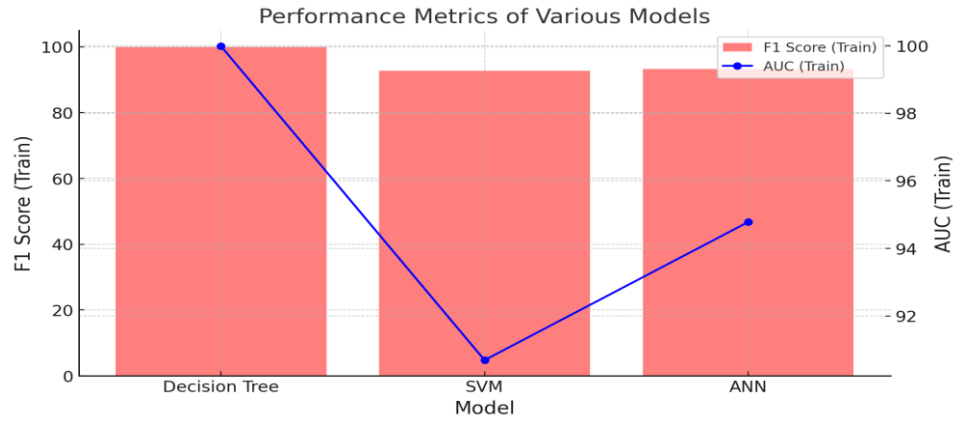

Figure 6: Train Performance Metrics of Various Models

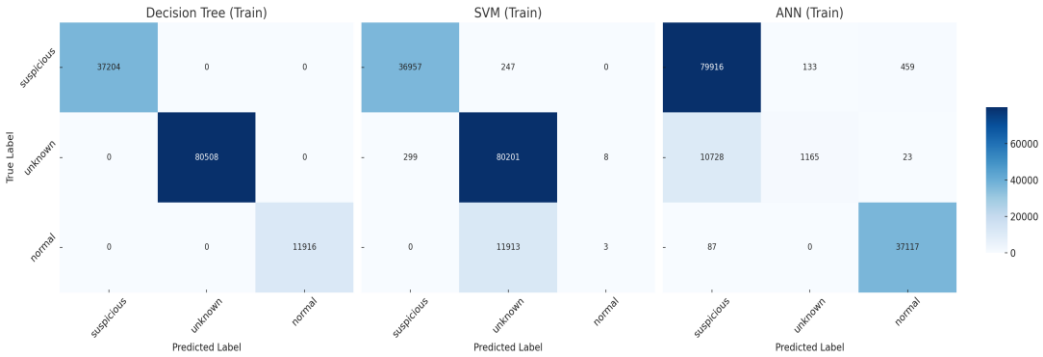

Figure 7: Confusion Matrix for Training

Testing Phase Results:

Table 2: Testing Phase Results

Model	**AUC (Testing) %**	**F1 Score (Testing) %**
Decision Tree	**99.93%**	**99.96%**
SVM	**90.57%**	**92.76%**
ANN	**94.78%**	**93.33%**

DOI: https://doi.org/10.54216/JCIM.140102
Received: January 06, 2024 Revised: Marach 18, 2024 Accepted: May 19, 2024

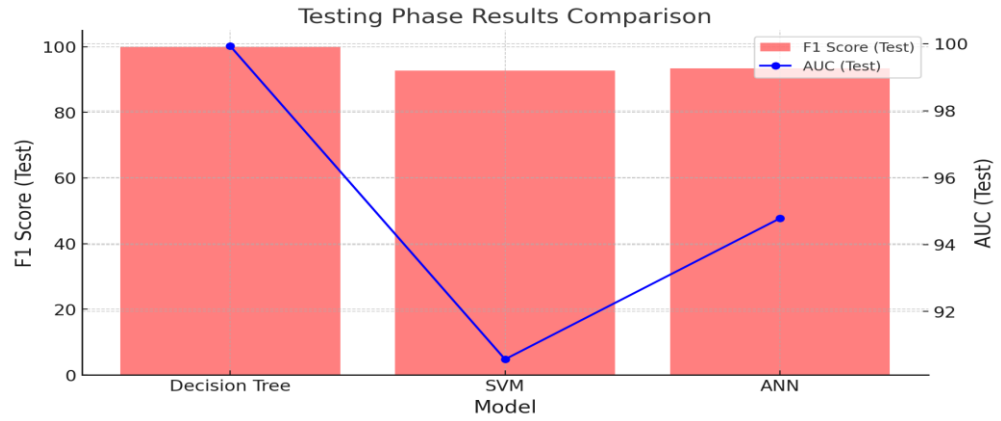

Figure 8: Testing Performance Metrics of Various Models

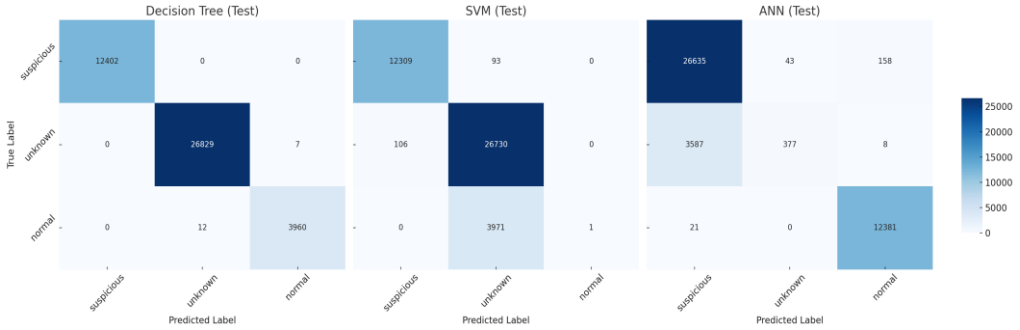

Figure 9: Confusion Matrix for Testing

As seen, the Decision Tree worked very well in the training phase of 99.99% F1 Score and AUC, and this was sustained during the testing phase with an F1 Score of 99.96% and an AUC of 99.93%. Its confusion matrix during the training phase was such that it allowed for perfect classification among all classes, and since this only fell by a negligible margin during the testing phase, it remained superior to the other models.

For the SVM, the training phase was an F1 Score of 92.79% and an AUC of 90.69%, decreasing a small fraction during the test phase to an F1 Score at 92.76% and an AUC at 90.57%. The confusion matrix derived from the training data showed a very strong ability for the model to correctly recognize most of the instances, and with a minimal number of misclassifications, and this was preserved during the test phase.

The ANN, in the same way close in the training phase, gives the F1 Score of 93.34 and AUC of approximately 94.78 during the test phase with F1 Score of 93.33 and AUC of 94.78%. The training confounding matrix denotes some misclassification; however, the model still shows a big capacity for correct classification. This was reflected in the testing phase results.

The following describes the relative strengths and weaknesses of each model based on the performance results. While the Decision Tree model gave the highest indicators on accuracy, the other two, SVM and ANN, were good models themselves and thus can each be used by preference, according to the specific requirements and constraints of any network traffic data classification task.

5. Discussion

The results of the study apparently unfold a complex landscape on the applications of machine learning in the domain of intrusion detection, which keeps getting shaped by new research like our own. This closely corresponds to the results of Alqahtani et al. for the year 2020, which state that it was well proven through the obtained results of our work that the model Decision Tree demonstrated high ability to classifying network traffic, with nearly perfect F1 and AUC scores, at stages of training and testing. Such fact supports the idea that tree-based algorithms remain quite a potent tool for cyber security tasks. Alqahtani by Alqahtani et al. mentioned the Random Forest algorithm for its accuracy,

DOI: https://doi.org/10.54216/JCIM.140102
Received: January 06, 2024 Revised: Marach 18, 2024 Accepted: May 19, 2024

which has similar methodological fundaments with the Decision Trees. Since besides, classification techniques further cement the strength of this paradigm. For example, in their work, Alzahrani et al. achieved high accuracy using ensemble methods, just as it is in the case of high accuracy of our Decision Tree. Our study more reports the confrontations by different performance metrics, giving a more granular insight into model efficacy.

Discovery by Rokade et al. and Sangkatsanee et al. from 2021 further follows our results on the Decision Tree performance, given the need for real-time systems for intrusion detection. Our study reports that Decision Trees are not just highly accurate, but at the same time, they maintain this level of performance in real-time application—highly effective in one detection. In contrast to Raghuvanshi et al., from 2022, who results showed the SVM to be more accurate, by their results, the SVM is not placed on top overall, but behind the Decision Tree. This possibly gives the hint that the performance of the SVM is very dataset-sensitive, which most probably depends on the dataset's processing or the feature-selection differences.

This high precision reported by Faker and Dogdou for deep learning models is in line with our results for ANN, especially for multiclass classification cases. Our contribution to the debate is that, in contrast, it shows that ANNs are powerful while being trumped by more traditional models, like Decision Trees, based on the complexity of the task and the dataset. Shin and Kim's work, focused on SVM, KNN, and Logistic Regression, and the reported accuracies, offered an important benchmark for our work. Our results suggest that, while these models are effective, they might not always propose the best performance, like in our case with the Decision Tree.

The works of Barbhuiya et al. (2020) and Alameidy et al. (2019) show the effectiveness of machine learning under dynamic environments and the importance of feature selection to model accuracy. We stick to this, and through our results prove that careful feature selection can indeed lead to great improvement in the performance of the model. The way that Siganos et al. research the integration of machine learning with the necessity for the models to be explained underlines an emerging issue of these models, which should not only be accurate but also transparent. Our study contributes, as we provide a clear and detailed analysis of model predictions through confusion matrices.

We further show our results by stressing the focus of Espes et al. on precision, where false positives have to be minimized in a correctly balanced precision-recall trade-off, probably the most delicate part of the IDS domain. Our great use of the F1 Score as an evaluation metric corroborates our models aligned with the precision-focused researches in the field.

6. Conclusion

This research has well proven the use of machine learning techniques on intrusion detection, especially on performance comparison using the models Decision Tree, SVM, and ANN. It was seen that the rank of performance vis-à-vis other models, from different key metrics such as the F1 Score and AUC, lies with the Decision Tree model in both the training and the test phases. It was also noted throughout the study that the selection of models would be based on the properties of the data and processing techniques. Although the Decision Trees gave better accuracy and efficiency, the performance of the SVM and ANN models implied that they might be applied in situations in which attributes of other models would be preferable.

Discussion of the use of machine learning in cybersecurity, as studied here, is really well-aligned within the current state of knowledge but also adds to the same by adding detailed comparative analysis and practical case studies. The future work will mainly build on the present results, resulting in hybrid models or applying deeper deep learning approaches.

Acknowledgements

I would like to just extend my appreciation to all the people who supported me during this research project. Special thanks will be directed to my supervisor, Dr. Hanan AlShaher, who had great experience and guided me in shaping the research and analysis. I appreciate my colleagues and the staff of the department for being of enormous help to me with their encouraging insights.I extend my thanks to all the participants and organizations that contributed data and resources, facilitating a comprehensive study.,Finally, I am thankful to my family and friends for their unwavering support and patience throughout the duration of this project.

DOI: https://doi.org/10.54216/JCIM.140102
Received: January 06, 2024 Revised: Marach 18, 2024 Accepted: May 19, 2024

References

[1] Sarker, I.H., et al. (2020). Cybersecurity data science: an overview from machine learning perspective.

[2] Tapiador, J.E., Orfila, A., Ribagorda, A., & Ramos, B. (2013). Key recovery attacks on KIDS, a keyed anomaly detection system. IEEE Transactions on Dependable and Secure Computing, 12(3), 312-325.

[3] Tavallaee, M., Stakhanova, N., & Ghorbani, A.A. (2010). Toward credible evaluation of anomaly-based intrusion-detection methods. IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews, 40(5), 516-524.

[4] Mohammadi, S., Mirvaziri, H., Ghazizadeh-Ahsaee, M., & Karimipour, H. (2019). Cyber intrusion detection by combined feature selection algorithm. Journal of Information Security and Applications, 44, 80-88.

[5] Maseer, Z.K., Yusof, R., Bahaman, N., Mostafa, S.A., & Foozy, C.F.M. (2021). Benchmarking of machine learning for anomaly based intrusion detection systems in the CICIDS2017 dataset. IEEE Access, 9, 22351-22370.

[6] Mishra, P., Varadharajan, V., Tupakula, U., & Pilli, E.S. (2018). A detailed investigation and analysis of using machine learning techniques for intrusion detection. IEEE Communications Surveys & Tutorials, 21(1), 686-728.

[7] Ambarwari, A., Adrian, Q.J., & Herdiyeni, Y. (2020). Analysis of the effect of data scaling on the performance of machine learning algorithms for plant identification. Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi), 4(1), 117-122.

[8] Liu, H., & Lang, B. (2019). Machine learning and deep learning methods for intrusion detection systems: A survey. Applied Sciences, 9(20), 4396.

[9] Dalinina. (2017). Introduction to Forecasting with ARIMA in R. Retrieved from https://www.datascience.com/blog/introduction-to-forecasting-with-arima-in-r-learn-datascience-tutorials

[10] Alqahtani, H., Sarker, I.H., Kalim, A., Hossain, M., Md, S., Ikhlaq, S., & Hossain, S. (2020, March). Cyber intrusion detection using machine learning classification techniques. In International Conference on Computing Science, Communication and Security (pp. 121-131). Springer, Singapore.

[11] Alzahrani, A.O., & Alenazi, M.J. (2021). Designing a network intrusion detection system based on machine learning for software defined networks. Future Internet, 13(5), 111.

[12] Rokade, M.D., & Sharma, Y.K. (2021, March). MLIDS: A Machine Learning Approach for Intrusion Detection for Real Time Network Dataset. In 2021 International Conference on Emerging Smart Computing and Informatics (ESCI) (pp. 533-536). IEEE.

[13] Sangkatsanee, P., Wattanapongsakorn, N., & Charnsripinyo, C. (2021). Practical real-time intrusion detection using machine learning approaches. Computer Communications, 34(18), 2227-2235.

[14] Raghuvanshi, A., Singh, U.K., Sajja, G.S., Pallathadka, H., Asenso, E., Kamal, M., & Phasinam, K. (2022). Intrusion detection using machine learning for risk mitigation in IoT-enabled smart irrigation in smart farming. Journal of Food Quality, 2022.

[15] Faker, O., & Dogdu, E. (2019, April). Intrusion detection using big data and deep learning techniques. In Proceedings of the 2019 ACM Southeast Conference (pp. 86-93).

[16] Shin, Y., & Kim, K. (2020). Comparison of anomaly detection accuracy of host-based intrusion detection systems based on different machine learning algorithms. International Journal of Advanced Computer Science and Applications, 11(2). http://dx.doi.org/10.14569/IJACSA.2020.0110233

[17] Wester, P., Heiding, F., & Lagerström, R. (2021). Anomaly-based intrusion detection using tree augmented naive Bayes. In Proceedings of the 2021 IEEE 25th International Enterprise Distributed Object Computing Workshop (EDOCW). IEEE. https://doi.org/10.1109/EDOCW52865.2021.00040

[18] Barbhuiya, S., Kilpatrick, P., & Nikolopoulos, D. S. (2020). DroidLight: Lightweight Anomaly-based Intrusion Detection System for Smartphone Devices. In Proceedings of the 21st International Conference on Distributed Computing and Networking (ICDCN 2020), January 4–7, Kolkata, India. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3369740.3369796

DOI: https://doi.org/10.54216/JCIM.140102
Received: January 06, 2024 Revised: Marach 18, 2024 Accepted: May 19, 2024

[19] Alamiedy, T.A., Anbar, M., Alqattan, Z.N.M. et al. Anomaly-based intrusion detection system using multi-objective grey wolf optimisation algorithm. J Ambient Intell Human Comput 11, 3735–3756 (2020). https://doi.org/10.1007/s12652-019-01569-8

[20] Yihunie, F., Abdelfattah, E., & Regmi, A. (2019). Applying Machine Learning to Anomaly-Based Intrusion Detection Systems. 2019 IEEE Long Island Systems, Applications and Technology Conference (LISAT), Farmingdale, NY, USA, 2019, pp. 1-5. https://doi.org/10.1109/LISAT.2019.8817340

[21] Bhavsar, M., Roy, K., Kelly, J., & Olusola, O. (2023). Anomaly-based intrusion detection system for IoT applications. Discover Internet of Things, 3(5). https://doi.org/10.1007/s43926-023-00034-5

[22] Siganos, M., Radoglou-Grammatikis, P., Kotsiuba, I., Markakis, E., Moscholios, I., Goudos, S., & Sarigiannidis, P. (2023). Explainable AI-based intrusion detection in the Internet of Things. In The 18th International Conference on Availability, Reliability and Security (ARES 2023), August 29–September 01, 2023, Benevento, Italy. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3600160.3605162

[23] Safavian, S.R., & Landgrebe, D. (1991). A survey of decision tree classifier methodology. IEEE Transactions on Systems, Man, and Cybernetics, 21(3), 660-674.

[24] González, L.A., Bishop-Hurley, G.J., Handcock, R.N., & Crossman, C. (2015). Behavioral classification of data from collars containing motion sensors in grazing cattle. Computers and Electronics in Agriculture, 110, 91-102.

[25] Charbuty, B., & Abdulazeez, A. (2021). Classification based on decision tree algorithm for machine learning. Journal of Applied Science and Technology Trends, 2(01), 20-28.

[26] Nashif, S., Raihan, M.R., Islam, M.R., & Imam, M.H. (2018). Heart disease detection by using machine learning algorithms and a real-time cardiovascular health monitoring system. World Journal of Engineering and Technology, 6(4), 854-873.

[27] Mohammadi, M., Rashid, T.A., Karim, S.H.T., Aldalwie, A.H.M., Tho, Q.T., Bidaki, M., & Hosseinzadeh, M. (2021). A comprehensive survey and taxonomy of the SVM-based intrusion detection systems. Journal of Network and Computer Applications, 178, 102983.

[28] Sheykhmousa, M., Mahdianpari, M., Ghanbari, H., Mohammadimanesh, F., Ghamisi, P., & Homayouni, S. (2020). Support vector machine versus random forest for remote sensing image classification: A meta-analysis and systematic review. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 13, 6308-6325.

[29] Gujral, K., Scott, J.Y., Ambady, L., Dismuke-Greer, C.E., Jacobs, J., Chow, A., & Yoon, J. (2022). A Primary Care Telehealth Pilot Program to Improve Access: Associations with Patients' Health Care Utilization and Costs. Telemedicine and e-Health, 28(5), 643-653.

DOI: https://doi.org/10.54216/JCIM.140102
Received: January 06, 2024 Revised: Marach 18, 2024 Accepted: May 19, 2024