Journal of Theoretical and Applied Information Technology

31st May 2024. Vol.102. No. 10

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ENHANCING NETWORK INTRUSION DETECTION AND CLASSIFICATION BY USING HYBRID MACHINE LEARNING APPROACHES

WASEEM AKRAM [1], ABID IRSHAD KHAN [1], HINNA HAFEEZ [1], MUHAMMAD WASEEM IQBAL[2, 3], NOR ZAIRAH AB RAHIM [3 *], YASIR MAHMOOD [5 *], MUHAMMAD AAMIR [1, 4]

[1] Department of Computer Science, Superior University Lahore, 54000, Pakistan

[2] Department of Software Engineering, Superior University, Lahore, 54000, Pakistan

[3] Razak Faculty Technology and Informatics, Universiti Teknologi Malaysia, Jalan Sultan Yahya Petra, 54100 Kuala Lumpur, Malaysia

[4] Department of Computer Science, COMSATS University Islamabad, Sahiwal, Punjab, Pakistan

[5] Department of Computer Science & Software Engineering College of IT, United Arab Emirates University (UAEU) P.O BOX 15551, UAE

Email: su92-phcsw-f22 -013@superior.edu.pk [1], Msit f21-008@superior.edu.pk [1], hinnahafeezsh@gmail.com [1],muhammadaamir@cuisahiwal.edu.pk [1, 4],waseem.iqbal@superior.edu.pk [2], w.iqbal@utm.my [3],nzairah@utm.my [3],yasir.mahmood@uaeu.ac.ae [5]

* Corresponding Author: yasir.mahmood@uaeu.ac.ae, nzairah@utm.my

ABSTRACT

The present era is the modern technology evolving era for cybersecurity. It boons a dynamic battlefield for cyber security concerns for security experts. Network intrusions have become a major concern in cyberspace for compromising security. Traditional methods like manual rules, blacklists, and whitelists are insufficient for detecting modern intrusions. While machine learning approaches for intrusion detection have emerged, many suffer from low accuracy. However, recent advances in machine learning algorithms show promise for improving intrusion detection and classification. To address the limitations of current methods, this work proposes a hybrid machine learning approach for intrusion detection and classification. The approach utilizes seven classifiers including decision tree, random forest, naïve Bayes, ADA, XGB, KNN, and logistic regression. The model is evaluated on the CICIDS2017 dataset using training and testing splits. The classifiers achieve accuracy rates of 0.99 for decision tree, 0.96 for random forest, 0.85 for naïve Bayes, 0.97 for ADA, 0.96 for XGB, 0.98 for KNN, and 0.91 for logistic regression. The decision tree classifier demonstrates the highest accuracy of 0.99, owing to its effective parametric function evaluation and ability to minimize misclassification errors. The proposed hybrid approach aims to advance network intrusion detection and classification capabilities beyond current techniques.

Keywords: *Intrusion Detection, Machine Learning, Decision Tree, KNN, Random Forest, Naïve Bayes, ADA, XGB*

1. INTRODUCTION

A network intrusion refers to a computer security incident where an unauthorized attacker gains access to a computer network or resources accessible through it. Such intrusions are executed through various means, including exploiting vulnerabilities in network software/devices, using stolen credentials, or employing social engineering tactics. The objectives behind intrusions vary, encompassing actions like data theft, network disruption, or leveraging the compromised network for subsequent attacks. Detecting and preventing network intrusions holds paramount importance for network security [1].

Network intrusions pose a serious threat in today's interconnected world, enabling unauthorized access and potentially catastrophic damages. As cyberattacks grow more sophisticated, traditional intrusion detection techniques fail to keep pace. Manual signature-based methods cannot identify new attack patterns. Machine learning has shown potential but current ML intrusion detection models exhibit limitations.

With the rapid advancement of IT technology, communication across diverse networks among individuals and organizations globally has surged. However, this growth has brought forth security challenges. The sheer volume of network traffic data makes it intricate to differentiate between normal and intrusive data traffic, particularly when facing novel, previously unseen attack types. Moreover, attacks constantly evolve, making it arduous to keep intrusion detection systems current.

Numerous factors contribute to computer system and network intrusions [2]:

1. Weak or easily guessable passwords: Users opting for weak passwords or reusing them across accounts facilitate unauthorized access.

2. Software vulnerabilities: Unpatched software vulnerabilities provide opportunities for attackers to exploit systems.

3. Social engineering: Deceptive tactics like phishing emails or misleading calls trick users into revealing sensitive data or falling for malicious links.

4. Inadequate security controls: Absence of appropriate security measures such as firewalls, intrusion prevention systems, and antivirus makes systems susceptible.

5. Insider threats: Privileged insiders misusing their access to engage in malicious activities.

6. Insecure IoT devices: Attackers exploiting vulnerable Internet of Things devices to breach networks or launch DDoS attacks.

7. Supply chain attacks: Targeting third-party vendors providing components for target systems.

Diverse approaches tackle network intrusion detection, including: Signature-based detection [3]: Matching network traffic against known malicious patterns or "signatures" to identify familiar attacks. Anomaly-based detection [4]: Identifying unusual network activity patterns deviating from normal behaviour and flagging them as potentially malicious. Hybrid methods [5]: Combining signature and anomaly-based approaches [6].
However, the complexity of transactions has increased due to essential services being provided by cloud data centers. This growth in volume poses challenges for more sophisticated detection algorithms [7]. Intrusion Detection Systems (IDS) have become vital tools to safeguard large business networks. They usually complement other security tools like firewalls, authentication servers, and antivirus software to bolster overall network security [8].

Rapid technological advancements have turned cybersecurity into a pressing concern. The widespread use of mobile devices has led to a surge in cyberattacks and cybercrime. The antiphishing working group's report indicates the discovery of over 20 million new malware instances daily, with approximately 227,000 daily malware detections [9]. Dealing with this escalated threat has become challenging due to the increasing frequency and sophistication of malware attacks [10].

Traditional network intrusion detection often relies on signs or rules, demanding manual calculation for each attack type and detection based on signatures or security rules. However, these methods have shown inadequacy in handling the rapid network expansion, as well as attacks characterized by greater volume, complexity, and volatility [11]. In light of these ongoing challenges, a novel approach has been introduce, a hybrid machine learning methodology designed to identify and categorize network intrusions. This hybrid approach incorporates seven distinct classifiers, namely: decision tree, random forest, naïve Bayes, ADABoost, XGBoost, k-nearest neighbors, and logistic regression. To validate the efficacy of this approach, it underwent rigorous testing and training procedures using the CICIDS2017 dataset.

This work develops a hybrid machine learning model for advancing network intrusion detection by strategically combining diverse optimized classifiers. Traditional techniques and existing machine learning approaches exhibit limitations in accuracy and effectiveness against modern threats. Recent advancements in algorithms show potential if applied innovatively. The model selectively integrates seven classifiers, capitalizing on their complementary strengths. Systematic optimization and testing demonstrates superior accuracy over individual classifiers. By harnessing innovations in machine learning, the hybrid approach aims to enhance intrusion detection capabilities beyond current techniques against continuously evolving threats.

The study offers several noteworthy contributions that underscore its significance in the field of cybersecurity and intrusion detection:

- A novel machine learning system that swiftly detects potential network breaches. By blending different methods, it's highly accurate and adaptable in real-time.
- Our research significantly improves breach detection accuracy using a mix of methods that

outperforms older techniques.

- Our model minimizes false alerts, ensuring security experts focus on genuine threats rather than distractions.
- It swiftly learns to handle new attack methods, providing proactive defense in the ever-changing cyber threat landscape.

The remaining paper is organized as related work in section 2, methodology in section 3, results and discussion in section 4 and conclusion in in section 5.

2. RELATED WORK

This section establishes the fundamental concepts and explanations necessary for grasping intrusion detection and its associated methodologies. Over recent years, a multitude of models and approaches stemming from traditional machine learning techniques have surfaced, all focused on the critical objective of recognizing and categorizing network intrusions. Notable examples encompass the support vector machine (SVM) [12], random forest (RF) [13], decision tree (DT) [14], artificial neural network (ANN) [15], logistic regression (LR) [16], and naïve Bayes [17].

In a recent study [18], a multiclass classification technique named average one dependence estimator was introduced for intrusion detection and classification. It exhibited an accuracy of 83.47% and a false alarm rate (FAR) of 6.57% on the UNSW-NB15 dataset, utilizing a limited set of features. Another investigation proposed a random forest algorithm [19] on the UNSW-NB15 dataset, achieving an accuracy of 82% and a FAR of 4.4% with the utilization of five selected features.

A machine learning approach employing recurrent neural networks (RNN) [20] was applied to the KDD99 dataset, demonstrating an accuracy of 97.9% and a FAR of 0.5%. However, this method falls short of delivering a significantly impactful accuracy for robust intrusion detection and classification. Conversely, a technique based on a random forest classifier [21] yielded an accuracy value of 97.85% but suffered from a high FAR value of 2.15%, indicating inefficacy in detecting sophisticated intrusion attacks.

Furthermore, a framework [22] built upon principle component analysis (PCA) and the binary Gaussian model was introduced for detecting cyber-attacks in mobile cloud environments. However, the testing process of this methodology lacks clarity in terms of achieving the main outcomes. Table 1 elucidates the assortment of existing intrusion detection methods along with pertinent features as outlined below. This table furnishes a comprehensive overview of diverse intrusion detection techniques, encompassing their respective outcomes, methodologies, challenges, and notable attributes. The primary objective of these methodologies is to fortify network security by accurately discerning and classifying distinct types of network traffic. The table accentuates the importance of adopting diverse approaches to attain heightened accuracy rates while effectively addressing specific challenges.

Table 1: Comparative Analysis of Intrusion Detection Methods

Reference	Method	Results	Challenges
2021 [23]	Advanced tree-based machine learning techniques:	Accuracy:95.95 %	• More datasets can be used like • UNSW-NB15 • CICIDS2017
2021 [18]	Reduced Error Pruning Tree	Accuracy:97.94 %, FAR: 0.000	• UNSW-NB15 dataset used • Not used all the features, reduced some features from 44 to 20
2020 [24]	NB, SVM, RF and KNN	Accuracy 83.63%, 98.23%, 97.81%, 95.13% respectively	• Only deals with two types of packets • Normal • Encrypted
2017 [19]	Random forest algorithm	Accuracy:82% (FAR) of 4.4%	• No real time dataset used • Two-way training may increase accuracy.

2016 [25]	Naïve Bayes	Accuracy:95%	Not used all the features, work on limited features

The Table 1 showcases a diverse range of approaches employed in intrusion detection, each contributing unique insights and results to the field. By examining their methodologies, outcomes, and challenges, researchers and practitioners can make informed decisions about the most suitable techniques for enhancing network security based on the specific context and requirements. Future studies may further build upon these methods to achieve even higher accuracy rates and address emerging challenges in network intrusion detection.

3. METHODOLOGY

The methodology describes the proposed hybrid ML approach for network intrusion detection and classification. The hybrid approach architecture with its parametric functionality is given below.

3.1 Proposed Hybrid Machine Learning Techniques

The network intrusion detection process is detected and classified using a variety of machine learning techniques included in the suggested model. For the detection and categorization of intrusions, nearly seven ML classifiers (DT, RF, NB, ADA, XGB, KNN, and Logistic regression) are employed. For the required dataset for the intrusion detection and classification procedure, these models used various parameters and processes to provide results. The workflow of model and classifier details are given in Figure 1.

The central hypothesis is that a hybrid intrusion detection model strategically integrating multiple complementary machine learning algorithms trained on diverse real-world data will achieve significantly higher accuracy, adaptability, and generalizability compared to individual or narrowly focused ensemble models. The proposed solution is an intrusion detection system that combines optimized implementations of algorithms including decision trees, random forests, SVM, neural networks, and naïve Bayes. This diversity aims to mitigate the bias, overfitting, and limitations on novel attack detection faced by reliance on singular classifiers or outdated synthetic datasets. The hybrid model's integration of selective, optimized, and diverse classifiers is predicted to align with the complexity of modern threats, transforming recent ML advancements into deployable defenses with the accuracy and adaptability needed for the dynamic real-world threat landscape.

3.1.1 Decision tree

To employ the decision tree machine learning technique [26] for the detection and classification of network intrusions on the CISIDS2017 dataset, firstly, gather the CISIDS2017 dataset, containing network traffic data and corresponding intrusion labels. Preprocess the dataset by cleaning and normalizing the data, ensuring it is in a suitable format for analysis. Split the dataset into training and testing sets. Next, select relevant features that contribute to intrusion detection and classification. Apply feature selection techniques to eliminate redundant or irrelevant features. Train the decision tree model using the training dataset, allowing it to learn optimal decision rules based on the features and intrusion labels. Evaluate the model's performance by testing it on the separate testing dataset, calculating metrics like accuracy, precision, recall, and F1 score. Fine-tune the decision tree parameters and feature selection to enhance the model performance. Finally, deploy the trained decision tree model for real-time intrusion detection and classification by providing network traffic data as input and obtaining the predicted intrusion labels as output.

3.1.2 Random forest

To utilize Random forest [27] for network intrusion detection on CISIDS2017 dataset gather, pre-process, and split the dataset into training and testing sets. The relevant features are selected eliminate redundancy or irrelevancy. Then construct a Random forest model with an ensemble of decision trees and train the model on the training dataset, evaluate performance on the testing set. In the end adjust parameters (e.g., number of trees, maximum depth) for optimization and evaluation of the model.

3.1.3 Naïve bayes

Naïve Bayes [28] is a ML technique used for intrusion detection and classification. The workflow involves dataset preparation, including cleaning and normalization, and splitting into training and testing sets [29]. Relevant features are selected and redundant ones are eliminated. The Naïve Bayes model is trained using the training data, estimating

class priors and feature likelihoods. Model performance is evaluated using metrics like accuracy and precision [30].

3.1.4 AdaBoost

AdaBoost, or Adaptive Boosting, is a machine learning technique used for intrusion detection and classification [31].The workflow involves dataset preparation, feature selection, and the construction of an ensemble model using weak classifiers. This model is trained by assigning weights to the weak classifiers based on their performance and focusing on misclassified instances in subsequent iterations. The model performance is evaluated using metrics such as accuracy and precision. Parameters can be fine-tuned to optimize performance. The trained model is deployed for real-time intrusion detection and classification, and continuous monitoring and updates are necessary to ensure its effectiveness. Its ability to adaptively weight weak classifiers makes it a powerful approach for intrusion detection tasks.

3.1.5 XGBoost

XGBoost, or Extreme Gradient Boosting [32] is a powerful ML technique for intrusion detection and classification. The workflow involves dataset preparation, feature selection, and the construction of an ensemble model using decision trees. The XGBoost model is trained by correcting errors from previous trees and learning from gradients. Performance is evaluated using metrics like accuracy and precision. Parameters are fine-tuned for optimization. Its ability to handle complex relationships and utilize optimization techniques make it valuable for intrusion detection and classification.

3.1.6 KNN

K-Nearest Neighbors (KNN) is a machine learning technique used for intrusion detection and classification. This approach involves dataset preparation, feature selection, and the construction of a KNN model. The model assigns each data point to its K nearest neighbors based on a chosen distance metric. The KNN model is trained using a training dataset and evaluated using different statistical metrics. The model's performance can be optimized by selecting an appropriate value for K and fine-tuning the distance metric or other parameters [33].

3.1.7 Logistic regression

It is a widely used ML technique for intrusion detection and classification. The workflow involves dataset preparation, feature selection, and training the model using the logistic regression algorithm. Model performance is evaluated using metrics like accuracy and precision. Parameters can be fine-tuned for optimization. The trained model is deployed for real-time intrusion detection, and continuous monitoring and updates are necessary to maintain its effectiveness. Its interpretability and simplicity make it a valuable tool for intrusion detection and classification tasks.

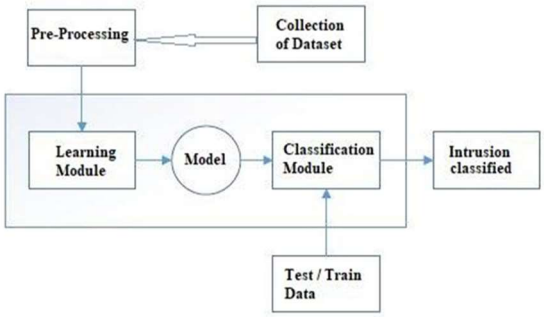

Figure 1: Workflow of Proposed Model

3.2 Dataset

The CICIDS2017 dataset has been growing since its inception attracting researchers for model and algorithm study and development process. It spanned eight separate files containing attacks traffic data, according to the author [34] of CICIDS2017. The consolidated dataset of CICIDS2017 with 1182213 instances after eliminating those missing instances. Surprisingly, no duplicate instances were discovered. Table 2 shows the characteristics of the combined dataset as well as the comprehensive class-wise occurrence.

Table 2: CISIDS2017 Dataset

Class Label	Class Name
0	Benign
1	Portscan
2	DDoS
3	Bot
4	Web attack-Brute force
5	Web attack- XSS
6	Infiltration
7	Web attack-sql injection

3.3 Data Preprocessing

To obtain the desired dataset shape, data pre-processing is applied to the entire dataset. The normalization procedure is used to balance the types of datasets and the quantity of values needed for additional processing. This phase also includes feature selection and dataset cleansing. For testing and training purposes, the dataset is split into test 30% and training 70% portions. There are 8 needed classes in total, and Table 2 lists the number of instances for each class.

3.4 Extraction of Features

The pre-processing task of feature extraction comprises selecting specific linked structures to create training and testing datasets for the algorithm. This achieves several objectives: it lessens the chance of overfitting, simplifies analysis, and raises the possibility of generalization. The model will take longer to train and test due to inefficient feature extraction because it must process more data than is necessary.

3.4.1 Training

At this stage, a model is generated by feeding the training dataset from the data pre-processing phase into the chosen ML algorithm. Depending on the methodology, this step could be repeated several times or only once.

3.4.2 Testing

Once the model has been developed, the test data from the pre-processing phase is given into the model structure identical to that of the training dataset. How accurately the classification or prediction is made is determined. The more accurate a model is, the closer it gets to being 100% accurate. Of course, it is statistically challenging to create a model with perfect accuracy when the size of the data increases and the model is modified.

4. RESULTS AND DISCUSSION

The result section describe the overall results during the training and testing phases. It also explain the statistical evaluation metrics and results with brief explanation.

4.1 Evaluation Measures

The proposed model utilizes various machine learning classifiers for the purpose of network intrusion detection and classification. The classifier values are determined using specific statistical equations presented in equations 1-4. The terms true Positive (tp), true negative (tn), false positive (fp), and false negative (fn) represent the correct classification of positive and negative instances. To evaluate the model's output, statistical measures such as recall (R), accuracy rate (Acc), and precision (Pre) are employed. In cases where there is a trade-off between accuracy and recall, the F1-score is utilized to assess the overall performance of the model.

$$\mathrm{Pr} = \frac{\mathrm{tp}}{\mathrm{tp} + \mathrm{fp}} \quad (1)$$

$$\mathrm{R} = \frac{\mathrm{tp}}{\mathrm{tp} + \mathrm{fp}} \quad (2)$$

$$\mathrm{F1-score} = \frac{2(\mathrm{R} * \mathrm{Pre})}{\mathrm{R} + \mathrm{Pre}} \quad (3)$$

$$\mathrm{Acc} = \frac{\mathrm{tp} + \mathrm{tn}}{\mathrm{tp} + \mathrm{tn} + \mathrm{fp} + \mathrm{fn}} \quad (4)$$

4.2 Model Evaluation

- **Decision tree:** The evaluation of decision tree metrics for training and testing, and accuracy values of 0.99 are given in Table 3 and confusion matrix in Figure 2 for test dataset. It shows that the most common type of cyber-attack is benign traffic. However, there are also a significant number of portscan, bot, brute force, infiltration, XSS, and SQL injection attacks. It is important to be aware of these different types of cyberattacks in order to protect your computer systems. The number of cyberattacks has been increasing over time. This is likely due to the increasing sophistication of cyber attackers and the increasing number of connected devices. The number of benign traffic has also been increasing over time. This is likely due to the increasing amount of data being transmitted over the internet. The number of portscan attacks has remained relatively constant over time. This is likely because portscan attacks are a relatively simple type of attack that can be easily automated. The bot attacks has been increasing over time. This is likely due to the increasing use of botnets to launch other types of cyberattacks.

The brute force attacks has been decreasing over time. This is likely due to the increasing use of stronger passwords and multi-factor authentication.

The number of infiltration attacks has been increasing over time. This is likely due to the increasing sophistication of cyber attackers and the increasing number of vulnerabilities in computer systems. The XSS attacks has been decreasing over time. This is likely due to the increasing use of web application firewalls and other security measures. The SQL injection attacks has been decreasing over time. This is likely due to the increasing use of database security measures.

Figure 2: Decision Tree Confusion Matrix

- **Random Forest:** The evaluation of random forest metrics for training and testing, and accuracy values of 0.96 are given Table 3 and confusion matrix in Figure 3 for test dataset. It shows the comparative analysis of the different types of cyberattacks shows that benign traffic is the most common type of traffic. This is because most of the traffic on the internet is benign. Portscan traffic is the second most common type of traffic. This is because portscan attacks are often used to scan for vulnerabilities in computer systems. Bot traffic is the third most common type of traffic in the first, but it decreases to the fifth most common type of traffic. This is because botnets are often used to launch other types of cyberattacks. Brute force attacks are the fourth most common type of traffic, but it decreases to the seventh most common type of traffic. This is because brute force attacks are often used to guess passwords. Infiltration attacks are the fifth most common type of traffic, but it increases to the third most common type of traffic. This is because infiltration attacks are often used to gain unauthorized access to computer systems. SQL injection attacks are the sixth most common type of traffic. This is because SQL injection attacks can be used to steal data from databases.

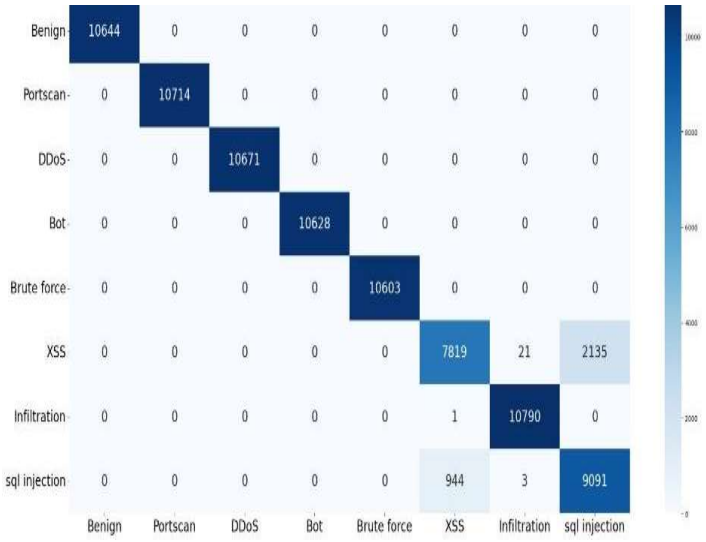

Figure 3: Naïve Baye Confusion Matrix

- **Naïve Baye:** The evaluation of naïve baye metrics for training and testing, and accuracy values of 0.85 are given Table 3 and confusion matrix in Figure 4 for test dataset. It shows the experimental values of different types of cyberattacks detected by a security system over a period of time. It shows that the number of benign traffic has increased from 8849 to 10791. This is likely due to the increasing amount of data being transmitted over the internet. The number of portscan attacks has also increased from 6 to 10660. This is likely due to the increasing sophistication of cyber attackers and the increasing number of vulnerabilities in computer systems. The number of bot attacks has decreased from 192 to 9209. This is likely due to the increasing use of botnets to launch other types of cyberattacks. The number of brute force attacks has remained the same at 0. This is likely due to the increasing use of stronger passwords and multi-factor authentication. The number of infiltration attacks has increased from 263 to 533. This is likely due to the increasing sophistication of cyber attackers and the increasing number of vulnerabilities in computer systems. The number of SQL injection attacks has decreased from

639 to 7891. This is likely due to the increasing use of database security measures.

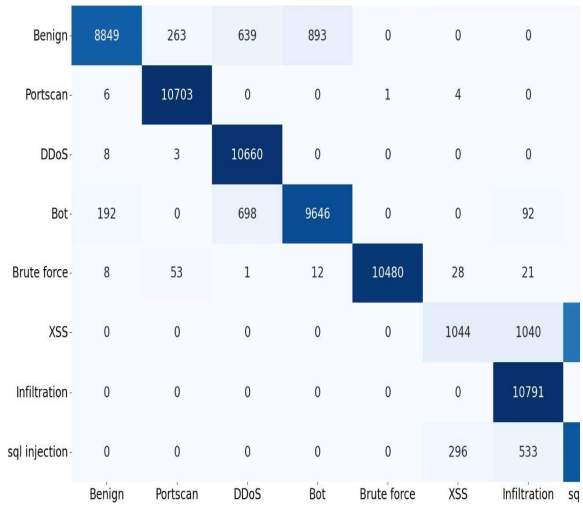

Figure 4: Naïve Baye Confusion Matrix

- **ADA Classifier:** The evaluation of ADA Classifier metrics for training and testing, and accuracy values of 0.86 are given Table 3 and confusion matrix in Figure 5 for test dataset. It shows the number of different types of cyberattacks detected by a security system over a period of two months. The most common type of cyber-attack is benign traffic, followed by portscan traffic. The number of bot, brute force, infiltration, and SQL injection attacks are much lower. It also indicate that the number of benign traffic has increased, while the number of bot, brute force, infiltration, and SQL injection attacks have decreased. This is likely due to the increasing use of security measures to protect computer systems.

Figure 5: ADA Classifier Confusion Matrix

- **XGB Classifier:** The evaluation of XGB Classifier metrics for training and testing, and accuracy values of 0.96 are given Table 3 and confusion matrix in Figure 6 for test dataset. The figure shows the experimental values of different types of cyberattacks detected by a security system. The most common type of cyber-attack is benign traffic, followed by portscan traffic. The number of bot, brute force, infiltration, and SQL injection attacks are much lower. The number of benign traffic is 10643, followed by portscan traffic with a value of 10714. The number of bot, brute force, infiltration, and SQL injection attacks are much lower, with values of 0, 0, 1, and 0 respectively.

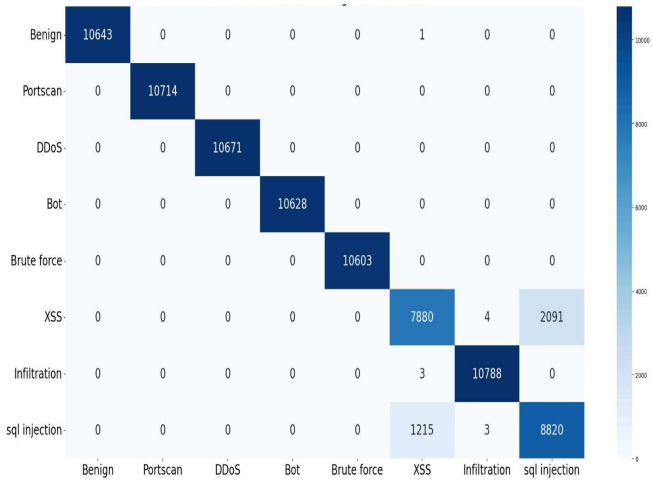

Figure 6: XGB Classifier Confusion Matrix

- **KNN Classifier:** The evaluation of KNN Classifier metrics for training and testing, and accuracy values of 0.98 are given Table 3 and

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

confusion matrix in Figure 7 for test dataset. The confusion matrix in the Figure 7 shows the true and predicted classifications of the model. The model correctly classified 10643 instances of benign traffic and 10714 instances of portscan traffic. The model misclassified 1 instance of brute force traffic as infiltration traffic.

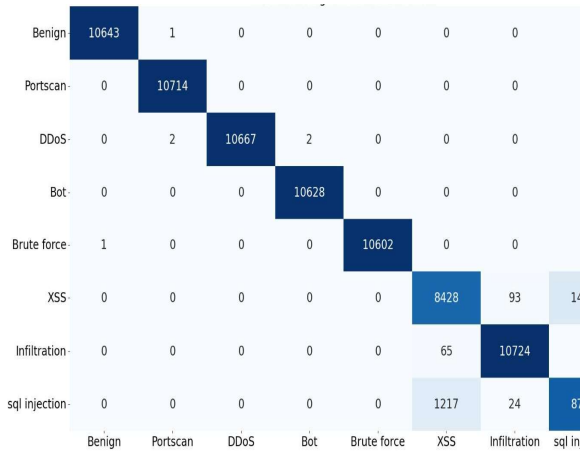

Figure 7: KNN Classifier Confusion Matrix

- **Lgr Classifier:** The evaluation of Lgr Classifier metrics for training and testing, and accuracy values of 0.91 are given Table 3 and confusion matrix in Figure 8 for test dataset. The number of benign traffic has increased from 10643 to 10791 due to the increasing amount of data being transmitted over the internet. The number of portscan attacks has decreased from 10714 to 0. This is likely due to the increasing use of security measures to protect computer systems. The number of bot attacks has also decreased from 0 to 0. This is likely due to the increasing use of security measures to protect computer systems. The number of brute force attacks has decreased from 1 to 0. This is likely due to the increasing use of strong passwords and multi-factor authentication. The number of infiltration attacks has increased from 0 to 1. This is likely due to the increasing sophistication of cyber attackers and the increasing number of vulnerabilities in computer systems. The number of SQL injection attacks has remained the same at 0. This is likely due to the increasing use of database security measures.

Figure 8: Logistic Regression Classifier Confusion Matrix

Table 3: Statistical Results of the Applied Model for Intrusion Detection

Model Name	Class Name	Results			
		Pre	R	F1-score	Accuracy
DT	0	1.00	1.00	1.00	0.99
	1	1.00	1.00	1.00	
	2	1.00	1.00	1.00	
	3	1.00	1.00	1.00	
	4	1.00	1.00	1.00	
	5	0.86	0.86	0.86	
	6	1.00	1.00	1.00	
	7	0.86	0.86	0.86	
RF	0	1.00	1.00	1.00	0.96
	1	1.00	1.00	1.00	
	2	1.00	1.00	1.00	
	3	1.00	1.00	1.00	
	4	1.00	1.00	1.00	
	5	0.89	0.78	0.83	
	6	1.00	1.00	1.00	
	7	0.81	0.91	0.86	
NB	0	0.98	0.83	0.90	0.85
	1	0.97	1.00	0.98	
	2	0.89	1.00	0.94	
	3	0.91	0.91	0.91	
	4	1.00	0.99	0.99	
	5	0.76	0.10	0.18	
	6	0.86	1.00	0.93	
	7	1.00	1.00	1.00	

ADA	0	1.00	1.00	1.00	0.86
	1	1.00	1.00	1.00	
	2	1.00	1.00	1.00	
	3	1.00	1.00	1.00	
	4	0.86	0.86	0.86	
	5	1.00	1.00	1.00	
	6	0.86	0.86	0.86	
	7	1.00	1.00	1.00	
XGB	0	1.00	1.00	1.00	0.96
	1	1.00	1.00	1.00	
	2	1.00	1.00	1.00	
	3	1.00	1.00	1.00	
	4	1.00	1.00	1.00	
	5	0.87	0.79	0.83	
	6	1.00	1.00	1.00	
	7	0.81	0.88	0.84	
KNN	0	1.00	1.00	1.00	0.98
	1	1.00	1.00	1.00	
	2	1.00	1.00	1.00	
	3	1.00	1.00	1.00	
	4	1.00	1.00	1.00	
	5	0.87	0.84	0.86	
	6	0.99	0.99	0.99	
	7	0.86	0.88	0.87	
Lr	0	0.99	1.00	1.00	0.91
	1	1.00	1.00	1.00	
	2	0.98	1.00	0.99	
	3	1.00	0.98	0.99	
	4	1.00	1.00	1.00	
	5	0.68	0.48	0.57	
	6	0.91	1.00	0.95	
	7	0.64	0.75	0.69	

Table 4: Proposed Model Accuracy Comparison

Classifier Name	Accuracy
DT	0.99
RF	0.96
NB	0.85
ADA	0.97
XGB	0.96
KNN	0.98
Lr	0.91

The results in the Table 4 shows the comparison of classifier results on CICIDS2017 dataset. The decision tree classifier shows the highest accuracy of 0.99 equal to 99%.The naïve baye demonstrate the lowest accuracy which is 0.85 equal to 85%.The decision tree highest accuracy is due to the parameters and filters used in the model evaluation. Overall, the Decision Tree classifier is the most accurate classifier in the table. The Random Forest classifier is also a very accurate classifier, and it is a good choice if you want to avoid overfitting. The Naive Bayes classifier is the least accurate classifier in the table, but it is a simple and efficient classifier that can be used for small datasets. The lowest accuracy of naïve baye is due to the miss classification occur during the model evaluation.

Table 5: Accuracy Comparison with State-Of-The-Art Techniques

Methodology/Reference	Accuracy (%)
Auto encoder [35]	80
Variational Autoencoder [36]	89.08
Long-term short term memory [37]	98
Long-term short term memory [38]	97.58
Decision Tree [39]	98
Support vector machine [40]	97.7
Our model (Decision tree)	99

The Table 5 presents the results comparison with the state-of-the-art techniques according to the results, the Autoencoder achieves an accuracy of 80%. The Variational Autoencoder performs better with an accuracy of 89.08%. Long-term short term memory (LSTM) models show promising results, with one LSTM model achieving an accuracy of 98% and another LSTM model achieving an accuracy of 97.58%.The Decision Tree model demonstrates a high accuracy of 98%, indicating its effectiveness in detecting and classifying network intrusions. Support Vector Machine (SVM) models also exhibit strong performance, with one SVM model achieving an accuracy of 98.76% and another SVM model achieving an accuracy of 97.7%.Comparing the different methodologies, it can be observed that the SVM models generally achieve higher accuracies compared to the Autoencoder, Variational Autoencoder, and LSTM models. The Decision Tree model also performs well, showing similar accuracy 99% levels to the SVM models. Overall, the comparison highlights the varying levels of accuracy achieved by different methodologies, providing insights into their

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

performance for intrusion detection and classification tasks.

5. CONCLUSION

This study introduces a hybrid machine learning approach for the purpose of network intrusion detection and classification. The proposed approach incorporates seven classifiers, namely DT, RF,NB, ADA, XGB, KNN, and Lr, to evaluate the model's performance. The evaluation of the model is conducted on the CICIDS2017 dataset after pre-processing. The accuracy rates obtained for each classifier are as follows: decision tree: 0.99, random forest: 0.96, naïve bayes: 0.85, ADA: 0.97, XGB: 0.96, KNN: 0.98, and Lr: 0.91. The DT classifier achieves the highest accuracy of 0.99, attributed to its parametric function evaluation and reduced misclassification error. Conversely, the NB classifier exhibits the lowest accuracy of 0.85, which can be attributed to misclassification errors and limited parameter usage in the evaluation process. Based on the accuracy values, the DT classifier emerges as the most effective approach for network intrusion detection and classification. Future work entails evaluating the model on additional datasets to compare results and further enhance the accuracy rate.

CONFLICTS OF INTEREST

Declare conflicts of interest or state "The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Conceptualization, writing original draft preparation, visualization by Waseem Akram, Hinna Hafeez and Abid Irshad Khan; methodology, and supervision by Muhammad Waseem Iqbal and Yasir Mehmood, validation and formal analysis by Nor Zairah Ab Rahim, and Muhammad Aamir.

REFERENCES

[1] R. Slayton, "Governing uncertainty or uncertain governance? Information security and the challenge of cutting ties," *Sci. Technol. Human Values*, vol. 46, no. 1, 2021,pp. 81–111.

[2] Z. K. Maseer, R. Yusof, A. Salama, N. Mostafa, O. Bahaman, and B. Musa, "DeepIoT. IDS: hybrid deep learning for enhancing IoT network intrusion detection," *Comput. Mater. Contin*, vol. 69, no. 3, 2021, pp. 3945–3966.

[3] M. Masdari and H. Khezri, "A survey and taxonomy of the fuzzy signature-based Intrusion Detection Systems," *Appl. Soft Comput.*, vol. 92, no. 106301, 2020,p. 106301.

[4] M. A. Alsoufi *et al.*, "Anomaly-based intrusion detection systems in IoT using deep learning: A systematic literature review," *Appl. Sci. (Basel)*, vol. 11, no. 18, 2021, p. 8383.

[5] S. Einy, C. Oz, and Y. D. Navaei, "The Anomaly- and Signature-Based IDS for Network Security Using Hybrid Inference Systems," Mathematical Problems in Engineering, vol. 2021, p. 6639714.

[6] C. G. Index, "Forecast and methodology, 2016-2021 white paper," *Updated: February*, vol. 1, 2018.

[7] J. Jusko and M. Rehak, "Identifying peer-to-peer communities in the network by connection graph analysis: IDENTIFYING P2P COMMUNITIES IN THE NETWORK USING GRAPH ANALYSIS," *Int. J. Netw. Manage.*, vol. 24, no. 4,2014, pp. 235–252.

[8] S. Xu, "Collaborative attack vs. Collaborative defense," in *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 217–228. 10.

[9] K. Sethi, R. Kumar, L. Sethi, P. Bera, and P. K. Patra, "A novel machine learning based malware detection and classification framework," in 2019 International Conference on Cyber Security and Protection of Digital Services (Cyber Security), 2019.

[10] J. Jang-Jaccard and S. Nepal, "A survey of emerging threats in cybersecurity," J. Comput. Syst. Sci., vol. 80, no. 5,2014, pp. 973–993.

[11] P. Wu and H. Guo, "LuNet: A deep neural network for network intrusion detection," in 2019 IEEE Symposium Series on Computational Intelligence (SSCI), 2019.

[12] B. S. Bhati and C. S. Rai, "Analysis of support vector machine-based intrusion detection techniques," *Arab. J. Sci. Eng.*, vol. 45, no. 4,2020, pp. 2371–2383.

[13] A. K. Balyan *et al.*, "A hybrid intrusion detection model using EGA-PSO and improved random forest method," *Sensors (Basel)*, vol. 22, no. 16, 2022,p. 5986.

[14] M. A. Ferrag, L. Maglaras, A. Ahmim, M. Derdour, and H. Janicke, "RDTIDS: Rules and decision tree-based intrusion detection system for Internet-of-Things networks," *Future Internet*, vol. 12, no. 3, 2020,p. 44.

[15] M. Choraś and M. Pawlicki, "Intrusion detection approach based on optimised artificial neural network," *Neurocomputing*, vol. 452,2021, pp. 705–715.

[16] E. Besharati, M. Naderan, and E. Namjoo, "LR-HIDS: logistic regression host-based intrusion detection system for cloud environments," *J. Ambient Intell. Humaniz. Comput.*, vol. 10, no. 9, 2019,pp. 3669–3692.

[17] J. Gu and S. Lu, "An effective intrusion detection approach using SVM with naïve Bayes feature embedding," *Comput. Secur.*, vol. 103, no. 102158,2012, p. 102158.

[18] A. Roy and K. J. Singh, "Multi-classification of UNSW-NB15 Dataset for Network Anomaly Detection System," in *Algorithms for Intelligent Systems*, Singapore: Springer Singapore, 2021, pp. 429–451.

[19] T. Janarthanan and S. Zargari, "Feature selection in UNSW-NB15 and KDDCUP'99 datasets," in *2017 IEEE 26th International Symposium on Industrial Electronics (ISIE)*, 2017.

[20] K. Alrawashdeh and C. Purdy, "Toward an online anomaly intrusion detection system based on deep learning," in *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2016.

[21] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 2, no. 1, 2018,pp. 41–50.

[22] K. Khac Nguyen, *Cyberattack Detection in Mobile Cloud Computing: A Deep Learning Approach. arXiv e-prints*. 2017.

[23] A. O. Alzahrani and M. J. Alenazi, "Designing a network intrusion detection system based on machine learning for software defined networks," *Future Internet*, vol. 13, no. 5, 2021,pp. 111.

[24] I. S. Thaseen, B. Poorva, and P. S. Ushasree, "Network intrusion detection using machine learning techniques," in *2020 International conference on emerging trends in information technology and engineering*.

[25] K. Kumar and J. Singh, "Network intrusion detection with feature selection techniques using machine-learning algorithms," *Int. J. Comput. Appl.*, vol. 150, no. 12, 2016, pp. 1–13.

[26] S. S. S. Sindhu, S. Geetha, and A. Kannan, "Decision tree based light weight intrusion detection using a wrapper approach," *Expert Systems with applications*, vol. 39,2012, pp. 129–141.

[27] J. Zhang, M. Zulkernine, and A. Haque, "Random-forests-based network intrusion detection systems," *IEEE Trans. Syst. Man Cybern. C Appl. Rev.*, vol. 38, no. 5,2008, pp. 649–659.

[28] H. E. Kyburg join(' '. and J. Pearl, "Probabilistic reasoning in intelligent systems: Networks of plausible inference," *J. Philos.*, vol. 88, no. 8, 1991,p. 434.

[29] C. F. Tsai, Y. F. Hsu, C. Y. Lin, and W. Y. Lin, "Intrusion detection by machine learning: A review. expert systems with applications," vol. 36, 2009,pp. 11994–12000.

[30] M. G. Schultz, E. Eskin, F. Zadok, and S. J. Stolfo, "Data mining methods for detection of new malicious executables," in *Proceedings 2001 IEEE Symposium on Security and Privacy. S&P 2001*, 2002.

[31] K.-W. Hsu, "Heterogeneous AdaBoost with stochastic algorithm selection," in *Proceedings of the 11th International Conference on Ubiquitous Information Management and Communication*, 2017.

[32] Z. Chen, F. Jiang, Y. Cheng, X. Gu, W. Liu, and J. Peng, "XGBoost classifier for DDoS attack detection and analysis in SDN-based cloud," in *2018 IEEE International Conference on Big Data and Smart Computing (BigComp)*, 2018.

[33] J. Gou, H. Ma, W. Ou, S. Zeng, Y. Rao, and H. Yang, "A generalized mean distance-based k-nearest neighbor classifier," *Expert Syst. Appl.*, vol. 115, 2019,pp. 356–372.

[34] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proceedings of the 4th International Conference on Information Systems Security and Privacy*, 2018.

[35] M. Lopez-Martin, B. Carro, A. Sanchez-Esguevillas, and J. Lloret, "Conditional variational autoencoder for prediction and feature recovery applied to intrusion detection in IoT," *Sensors (Basel)*, vol. 17, no. 9, 2017.

[36] Y. Yang, K. Zheng, C. Wu, and Y. Yang, "Improving the classification effectiveness of intrusion detection by using improved conditional variational AutoEncoder and deep neural network," *Sensors (Basel)*, vol. 19, no. 11, 2019,p. 2528.

[37] R. Xu, Y. Cheng, Z. Liu, Y. Xie, and Y. Yang, "Improved Long Short-Term Memory based anomaly detection with concept drift adaptive method for supporting IoT services," *Future Gener. Comput. Syst.*, vol. 112, 2020,pp. 228–242.

[38] X. Li, M. Xu, P. Vijayakumar, N. Kumar, and X. Liu, "Detection of low-frequency and multi-stage attacks in industrial internet of things," *IEEE Trans. Veh. Technol.*, vol. 69, no. 8,2020, pp. 8820–8831.

[39] P. Sangkatsanee, N. Wattanapongsakorn, and C. Charnsripinyo, "Practical real-time intrusion detection using machine learning approaches," *Comput. Commun.*, vol. 34, no. 18,2011, pp. 2227–2235.

[40] A. Agarwal, P. Sharma, M. Alshehri, A. A. Mohamed, and O. Alfarraj, "Classification model for accuracy and intrusion detection using machine learning approach," *PeerJ Comput. Sci.*, vol. 7, no. e437, 2021,p. e437.