Sharif et al. LGURJCSIT 2024

LGU Research Journal of
Computer Science & IT

ISSN: 2521-0122 (Online)
ISSN: 2519-7991 (Print)

doi: 10.54692/lgurjcsit.2024.082569

Vol (8): Issue (2), April – June 2024

A Machine Learning-Based Approach for the Detection of DDoS Attacks on the Internet of Things Using CICDDoS2019 Dataset – PortMap

Hanan Sharif[1], Sardar Usman[2], Muhammad Hasnain[3], Shagufta Anwar[4], Mohammed Nawaf Altouri[5], Fahad Mohammed Sharahili[6], M. Usman Ashraf [7*]

[1,3,4]Department of Computer Science, Leads University Lahore, Punjab, Pakistan.
[2]Department of Computer Science Software Engineering & IT, Grand Asian University, Sialkot, Punjab, Pakistan.
[5]University of prince muqrin, Madinah, Saudi Arabia.
[6]Imam mohammad Bn Saud Islamic University, Riyadh, Saudi Arabia.
[7]Department of Computer Science, GC Women University Sialkot, Punjab, Pakistan,

Email: usman.ashraf@gcwus.edu.pk

ABSTRACT:

In today's technological era, the Internet has become ubiquitous, playing a vital role in our daily lives. With the exponential growth of IoT innovation, millions of interconnected IoT-enabled devices rely on cloud services to communicate over the Internet. However, this rapid development also exposes these devices to various threats, with DDoS (Distributed Denial of Service) and DoS (Denial of Service) attacks being particularly potent and destructive. DDoS attacks present a unique challenge as they are tough to detect using conventional intrusion detection frameworks and traditional methodologies. Fortunately, advancements in machine learning have provided a promising solution by enabling accurate differentiation between DDoS attacks and other forms of data. This study proposes a DDoS detection model based on machine learning algorithms. We used the most recent and freely available online dataset called CICDDoS2019 to conduct this study. Various machine learning-based techniques were explored to identify the characteristics associated with accurate classification. Among the algorithms tested, AdaBoost and XGBoost demonstrated exceptional performance. A hybrid approach will be incorporated into this model as part of future work, further improving its capabilities. It is worth noting that this model will be continuously updated with new data on DDoS attacks, ensuring its relevance and effectiveness in combating emerging threats. By leveraging machine learning techniques, this approach enhances the detection of DDoS attacks on Internet of Things networks, safeguarding the integrity and security of connected devices and the overall IoT ecosystem.

KEYWORDS: DDoS, Internet of Things, Machine Learning, Classification, DDoS Detection, CICDDoS2019.

1. INTRODUCTION

The Internet of Things (IoT) continues revolutionizing our world, bringing numerous benefits and advancements. Today, IoT devices play a pivotal role in our daily lives, permeating various aspects such as smart cities, electricity grids, homes, vehicles, construction machinery, and hospitals. This exponential growth in digital technology aims to enhance our lives by seamlessly integrating physical devices with digital intelligence, creating a more comfortable, intelligent, and manageable environment. IoT devices collect vast amounts of data, which can be shared through the Internet, enabling access from anywhere at any time. These data streams are typically stored and accessed through integrated cloud platforms, facilitating communication among IoT devices. Research indicates that by 2030, the number of IoT devices is projected to reach 20 billion, with the current count already at 10.07 billion, all interconnected through the web [1]. However, with this extensive proliferation of interconnected devices comes the need to protect the data they generate. Cyber security is crucial to prevent unauthorized access and safeguard our valuable assets and personal privacy [2]. As the volume of data transferred among these devices continues to grow, robust security measures must be in place to mitigate the risk of cyber-attacks.

The cyber security threats faced by IoT devices can be classified into six types: denial of service, impersonation, eavesdropping, hardware tempering, bogus information, and message suspension. [3]. DoS and its more advanced version, DDoS, the abbreviation of distributed denial of service, are complicated and much more complex attacks to detect or mitigate than the other five types [4]. In this category of attack, a lot of information is sent through the servers, which brings about prevention of administration given by the specialist co-op. Due to this, consumers or users won't be able to use services properly and face problems in receiving proper service [5]. DDoS attacks are also classified into different types depending on different characteristics. A DDoS attack is classified into the following types [6]: 1) SYN Flood 2) TCP Flood 3) Ping of Death 4) DNS Flood 5) Zero-Day DDoS 6) HTTP Flood 7) ICMP Flood 8) SYN Flood 9) UDP Flood.

The primary objective of this study is to identify the most dependable, precise, and accurate algorithm for detecting DDoS attacks on IoT devices. The research utilizes the CICDDoS2019 dataset, depicted in Figure 1 (Model Architecture). The dataset is divided into two categories: "harmful" and "harmless" classes. This study introduces a machine learning model that rapidly detects DDoS attacks, comparing thoroughly with existing detection models. By utilizing the CICDDoS2019 dataset, the study refines dimension reduction and feature selection techniques for effective DDoS detection. Rigorous testing identifies the most suitable algorithm for detecting distributed denial of service attacks proficiently. The subsequent sections of the paper are structured as follows: Section II provides a concise introduction to the background, followed by a comprehensive literature review in Section III. Section IV outlines the intricate methodology employed in the study. The findings and their analysis are expounded upon in Section V. Finally, and Section VI encapsulates the conclusions drawn from the analysis and points towards potential future directions.

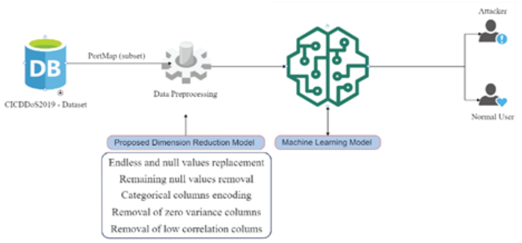

DB
CICDDoS2019 - Dataset
PortMap (subset)
Data Preprocessing
Attacker
Normal User
Proposed Dimension Reduction Model
Machine Learning Model
Endless and null values replacement
Remaining null values removal
Categorical columns encoding
Removal of zero variance columns
Removal of low correlation colums

Figure 1: Architectural framework of the proposed model

2. BACKGROUND

The term "IoT" stands for the Internet of Things, encompassing all internet-connected devices with specific functionalities in our daily lives. These devices include actuators, sensors, and microcontrollers. The IoT comprises countless physical, digital devices worldwide that collect, exchange, and analyze data. The applications of IoT are diverse and include connected communities, electric grids, electric mobility, smart homes, healthcare systems, and smart living, among others. [7]. Due to this extensive use and exponentially increasing data of IoT device users, it has become a significant threat to the privacy and usability of these devices. Many attacks have been focused on IoT gadgets since their development in the advanced world. Denial of Service and its advanced form, DDoS, are among today's most serious security threats. In this attack, the attacker aims to bypass an organization's resources by manipulating the incoming and outgoing network traffic. This leads to service disruptions and prevents authorized users from exercising their administrative rights effectively. DDoS attacks come in various forms, each with its consequences. It is crucial to clearly understand IoT and its functioning to distinguish between different types of attacks and their distinct characteristics. With diverse detection methods and mitigation strategies available for each type of DDoS attack, this knowledge becomes essential in effectively countering such threats.

Every web-enabled computer device that can transmit and receive data via sensors, communicate using a network, and have processing capability by using embedded processors is considered an IoT (Internet of Things). Existing and emerging technologies are utilized for sensing, network ing, and robotics. This allows the user to achieve deeper analysis, integration, and automation within a system. With the increase in networking capabilities of machines and appliances used in different fields daily, such as homes, offices, transportation, buildings, and industries, they open a world of opportunities for the betterment of business and better customer satisfaction. Some of the key features of the IoT are communication, sensors, artificial intelligence, small devices, and active engagement [8].

2.1. How IoT Works?

IoT technology is classified into four basic categories: IoT gadgets and devices, cloud-based data storage systems, remote controls used through mobile applications, and gateway systems. Combining systems can make connecting two or more devices possible [9]. In Fig. 2, we can see how the IoT layers are connected through a general diagram. The following are some key components of IoT technology that have a vital role in IoT device performance.

- **Sensor-based Technology:** Important information about gadgets can be detected from an extensive range of sensors attached to the devices. The collected information can be location, temperature, gases, any industrial machine's function, or sensory data for plant health [10].
- **IoT Gateways:** Gateways bridge the gap by providing a link between the end user and the IoT device, thus allowing them to connect and communicate [11].
- **Storage of Data and Cloud Server:** The cloud stores and analyses data. Collected data reaches the cloud after passing through the gateway. The data is then processed and transferred to the user for further proceedings. A user executes different actions on data depending on the achieved information [12].
- **Usage of Mobile Applications for Remote Controls:** Remote controls are used by end users through mobile phones, laptops, tablets, etc., and they have different applications installed on them. These applications control, monitor, retrieve data, and perform different actions on a user's remote-controlled IoT gadgets [13].

2.2. DDoS Attacks - Detection

There are a variety of strategies for successfully detecting DDoS attacks. However, the new confounded types of attacks make traditional ones increasingly difficult to distinguish. The most proficient method for distinguishing between these attacks is to utilize information mining and machine learning strategies. In these sorts of methods, a lot of information is gathered in a reproduced environment or genuine attack; then, at that point, analysts separate key features from crude information. From that point forward, ML-based methods are used to create the detection model, and the model's performance is evaluated to determine whether the method is appropriate for detecting DDoS attacks. A rundown of standard machine learning algorithms for DDoS detection is accessible as follows:

- **Random Forest:** This is a decision tree-based algorithm that is also used primarily for the classification of datasets and some other tasks, which are carried out by constructing many decision trees from a training set that was randomly selected. It combines the votes from numerous decision trees to determine the object's exact class. A separate loss for each class label for each observation is made for a multiclass classification.
- **AdaBoost:** This machine-learning approach works by assigning weights to observations. Cases are given weightage based on their identification. New weak learners are introduced sequentially, allowing them to focus on increasingly challenging patterns. Boosting has two primary challenges.
- How can the training set be adjusted so the weak classifier can learn from it?
- How do we make a strong classifier out of the poor classifiers created during training?

In 1995, the Adaboost (adaptive boosting) strategy to address these difficulties was introduced by Freund and Schapire, which worked by altering weightage without requiring any prior knowledge of learner learning. The algorithm can solve many of the early boost method's practical issues and adjust voting weights. [14].

- **XGBoost:** Xtreme Gradient Boosting is a technique that employs an iterative strategy to improve model accuracy by reducing errors. The augmentation of gradient boosted decision trees (GBM) provides adaptable, easy, and DGBL results (Distributed gradient boosting library). XGBoost may improve performance and efficiency for classification, regression, and parallel computing issues. It tends to be utilized as an implicit system with the assistance of SageMaker to accomplish better versatility and a way to deal with more further developed construction, for example, K-overlay cross approval, as you can alter your preparation scripts.

The three essential components of XGBoost are a loss function to assess model prediction percentage, a weak learner to categorize data while guessing incorrectly, and an additive function to reduce the loss function's value through repeated and sequential processing [15] [16].

- **Naïve Bayes:** This basic Bayes' Theorem-based probabilistic classifier works well with huge datasets. When the characteristics in the datasets are independent of one another, the Naive Bayes model is simple to construct. The classifier is quick and unaffected by irrelevant characteristics. In binary scenarios, such as when the goal of classification is to determine if arriving packets are DDoS or not, the Nave Bayes algorithm works exceptionally well. The model learns by computing the probability of training data. The naive Bayes classifier simplifies learning by assuming that attributes are independent of class. The class conditional mutual information between characteristics is defined as component dependence and does not affect naive Bayes accuracy. On the other hand, the amount of information about the class is lost because the independence assumption is a better predictor of the success of Naive Bayes classification [17].
- **SVM:** To solve the problems of regression and classification, a supervised ML approach is known as SVM (Support Vector Machine). However, it is often utilized in grading. A hyperplane in N-dimensional space is used to classify the data points (N is the number of characteristics). With roots in Statistical Learning Theory (SLT) and optimization methods, support vector machines have evolved into powerful ML problem-solving tools with finite training points, overcoming some traditional challenges, including over-fitting, the curse of dimensionality, and so on. Implementation techniques and theoretical foundations for SVMs have been established due to several appealing features, including good generalization abilities, enticing mathematical representations, geometrical explanations, and promising empirical performance. SVMs are gaining popularity and development at a rapid pace [18].
- **KNN:** It is a simple and easily applicable supervised ML-based algorithm that is best for problem-solving in regression and classification. The k-Nearest-Neighbors (kNN) technique is a basic yet successful classification method. In many cases, kNN is a basic yet effective non-parametric classification algorithm. To classify an x number of data records, the k closest neighbours are collected, forming a neighbourhood of x. The category of x is usually determined by a majority vote amongst collected data in the neighborhood, with or without distance-based weighting [19]. We use the distance metric to compare and find relativity between existing K examples of a training dataset and the upcoming input.

3. LITERATURE REVIEW

Jie Xue et al. [20] presented a model that utilized 7 features extracted from the user's application layer to distinguish between regular users and bots. Their primary objective was to identify and differentiate bot behaviour. They incorporated a one-class SVM algorithm into their collected database and concluded that their model effectively detected denial of service attacks at the application layer. Shrikhand Wankhede et al. [21] conducted an analysis and proposed a detection model based on machine learning. In addition to utilizing machine learning techniques, the researchers incorporated neural networks to enhance the accuracy of their model by optimizing a set number of parameters. They employed the 500-tree random forest algorithm to train their model on half of the dataset and achieved an impressive accuracy of 99.95 percent. It is important to note that an earlier version of the CIC IDS was utilized in this study.

Jakula et al. [22] researched detecting DDoS attacks using supervised and unsupervised machine learning approaches previously explored by other researchers. Their study introduced a novel approach for identifying DDoS attacks, incorporating a new parameter called P (A), which significantly contributed to their research. They enhanced their algorithm's accuracy and performance by determining the optimal number of hyperparameters. The parameter P (A) was utilized as a benchmark to make informed decisions during model training. Through experimentation on the "NSL-KDD" dataset, they found that certain algorithms such as Naive Bayes, Gradient Boost, and Random Forest yielded the best precision and training time results.

Dong et al. [19] introduced two innovative algorithms, DDADA and DDAML, based on the concepts of K-Nearest Neighbors (KNN) and attack intensity. They collected datasets from a simulation environment to generate DDoS traffic. Then they evaluated the performance of their proposed algorithms in comparison to traditional AI algorithms such as KNN, SVM, and Naïve

Bayes. By examining the ROC curve results, they discovered that their new algorithms outperformed the existing ones, showcasing improved detection capabilities for DDoS attacks.

Patil et al. [23] offered a DDoS location framework in light of solicitation package header connections. For testing purposes, researchers used the Caida dataset and the real separated information that utilized the ideas of modularity, SVM calculation, and entropy. Researchers concluded that higher precision can be achieved by adding the concept of entropy to UDP association and modularity to TCP association.

R. Boss et al. [24] contrasted various calculations for conventional learning and crossover strategies. They tried these calculations in the DARPF and KDDcup99 informational collections, observing that decision trees and C-Mean work well compared to other researchers' work. There is a 98.7% chance that the Fuzzy C-Mean calculation can tell if there is DDoS traffic with 0.15 seconds of identifying season.

Doshi et al. [25] were encouraged to develop new ways to automatically detect IoT consumer traffic attacks. Researchers showed that using IoT-specific network behaviour to guide feature selection can lead to the acquisition of accurate attack detection using a range of ML methods, which include neural networks. These findings suggest that using cheap machine-learning strategies and protocol-agnostic-based traffic data by home gateways or other internal network routers may automatically identify IoT device resources for DDoS attacks. Machine learning is extensively utilized to tackle various challenges in diverse domains [26-30]. The versatility of machine learning is evident in its application across fields such as cybersecurity, agriculture, energy management, and more.

This research aims to create a machine-learning model capable of swiftly and accurately identifying DDoS attacks. The study delves into optimizing dimension reduction and feature selection techniques for DDoS detection using the CICDDoS2019 dataset. Rigorous testing determines the optimal algorithm for proficient and efficient detection of DDoS attacks.

4. METHODOLOGY

4.1. Proposed Study

Initially, an optimal machine learning (ML) algorithm commonly employed for Dos/DDoS attack detection is identified through a comprehensive review of related works by various authors. Utilizing observed data, a model is formulated to assess the effectiveness and execution speed of various algorithms. The "CICDDoS2019" dataset is utilized for training and testing the models. The model encompasses multiple phases, with preprocessing involving the extraction of an effective feature set for model training. Subsequent testing evaluates the performance of different algorithms to determine the optimal solution. Ultimately, the study identifies crucial features within the CICDDoS2019 dataset that significantly influence accurate DDoS predictions.

4.2. Dataset Preparation

This research employs the latest available dataset, named CICDoS2019, for studying DDoS attacks, addressing the limitations of previous datasets. This dataset encompasses two types of DDoS attacks: reflection-based and exploitation-based, utilizing TCP/UDP protocols at the application layer. Notably, the dataset introduces a novel classification method with new attack types, which is a significant advantage. Diverse DDoS attack categories, including "WebDDoS," "SNMP," "NTP," "DNS," and more, are categorized, while regular traffic is labeled as "BENIGN." These labeled network traffic data and associated features are stored in an accessible CSV file. The traffic features are extracted using CICFlowMeter-V3.

4.3. Pre-processing of Data

Direct utilization of the CICDDoS2019 dataset for model training and testing is hindered due to its large size (around 3 GB), necessitating robust processing capabilities. Consequently, only the Portmap section of the dataset is employed, focusing on a streamlined set of essential features. Data preprocessing is conducted using Google Colab, a web-based Python coding environment developed by Google, which is highly regarded among data analysts and ML researchers. The preprocessing phase involves the utilization of various libraries, including Pandas, Numpy, and Scikit-learn, to accomplish tasks effectively.

4.4. Dimension Reduction of Dataset

The vital phases for proposed dimension reduction model are concluded as:

- **Replace Infinite and Null Values:** To prevent the mixing of SI and CGS units, like using amperes for current and oersteds for magnetic field, which often leads to dimensional imbalances in equations, it's essential to avoid such combinations. If mixed units are necessary, ensure clear unit definitions for each quantity in the equation.

- **Eliminate Remaining Null Values:** String-type null values in the dataset can disrupt experiments. Since these values can't be replaced, they are removed from the dataset, which contains a sufficient number of records.
- **Encode Categorical Columns:** As part of data preprocessing, convert all string values in the dataset into numerical values. This encoding ensures precise experimental outcomes.
- **Discard Zero Variance Columns:** Columns with zero variance are eliminated since they contain identical values, offering no impact on results.
- **Prune Low Correlation Columns:** Removing insignificant features is crucial for dimension reduction, preventing overfitting, and enhancing the model's execution speed. This step involves eliminating columns with low correlation.

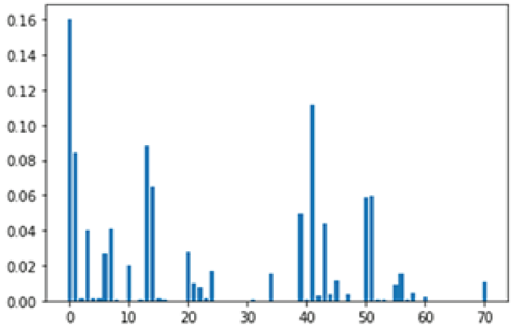


Figure 2: Complete Features and their Importance

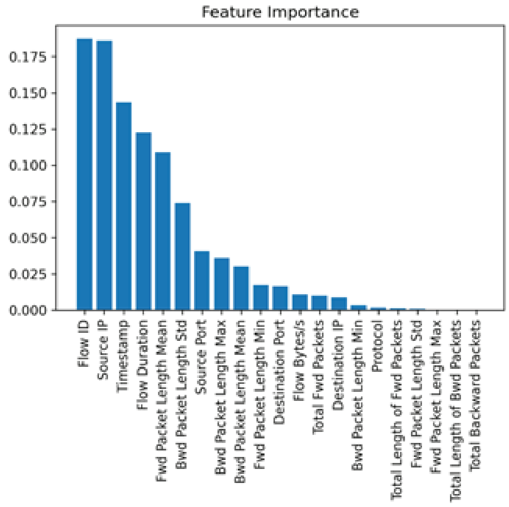


Figure 3: Top 20 Important Features

Figure 2 visually depicts the significance of all the features extracted from the dataset during the data preprocessing stage. Figure 3 presents the top 20 significant attributes influencing network class prediction. The X-axis delineates these key features, which hold sway over both the dataset and model performance. The Y-axis represents the proportional impact of each feature. Flow ID and Source IP are the most influential factors in DDoS attack detection systems.

4.5. Machine Learning Based Algorithms

Numerous studies by fellow researchers focusing on detecting DDoS attacks through machine learning techniques are explored. Prominent among these techniques are Random Forest, SVM, Naive Bayes, KNN, XGBoost, and AdaBoost—widely recognized ML algorithms known for their efficacy in identifying distributed denial of service attacks. The proposed model is subjected to training and testing utilizing these algorithms to determine the most proficient performer. Subsequent sections delve into the assessment metrics used for evaluation.

4.6. Evaluation

Several evaluation metrics are employed to compare the performance of ML algorithms and extract valuable features:

- **Classification Accuracy:** This measures the proportion of accurate predictions out of the total predictions made. However, solely relying on accuracy might be inadequate due to potential imbalances in the dataset.
- **F1-Score:** The F1-Score combines precision and recall into a single metric, offering a comprehensive assessment of false positives and false negatives. It is a more robust testing measure.
- **Training Time:** This metric gauges a model's efficiency and speed during training.
- **Feature Importance:** This assesses the correlation between each feature and its predicted labels, aiding in identifying influential attributes.

5. RESULTS AND ANALYSIS

This section uses various algorithms to present the outcomes of the comparative analysis between the proposed model and the CICDDoS2019 dataset. The results are meticulously examined to determine the optimal algorithm for DDoS attack detection. As depicted in Figure 4, the performance of all algorithms is evaluated, with Naïve Bayes being the exception, achieving an

F1-score of 0.6802, indicating true positive predictions. However, Naïve Bayes displays limitations in DDoS attack detection due to its elevated rate of false positives. This implies that Naïve Bayes tends to misclassify benign traffic as malicious, potentially hindering effective attack prevention.

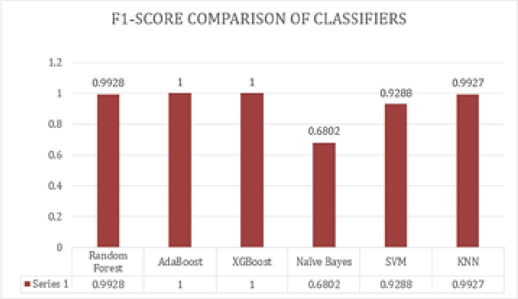
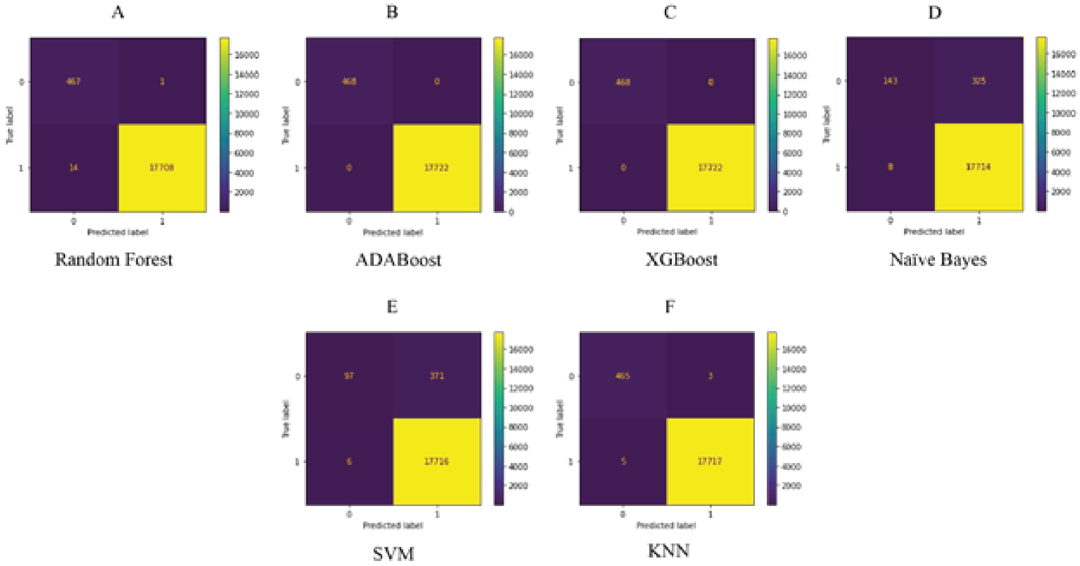


Figure 4: F1-Score Comparison of All Classifiers

In Figure 6, the outcomes reveal that Naïve Bayes excels in terms of training time, requiring a mere 0.414 seconds to complete model training. Following that, Random Forest concluded training in 4.645 seconds, while XGBoost and AdaBoost took 10.325 and 14.853 seconds, respectively. In contrast, KNN and SVM exhibited the lengthiest training times, consuming 97.795 and 1711.976 seconds, respectively.



Random Forest

ADABoost

XGBoost

Naïve Bayes

SVM

KNN

Fig.ure 5: Confusion Matrix of all algorithms

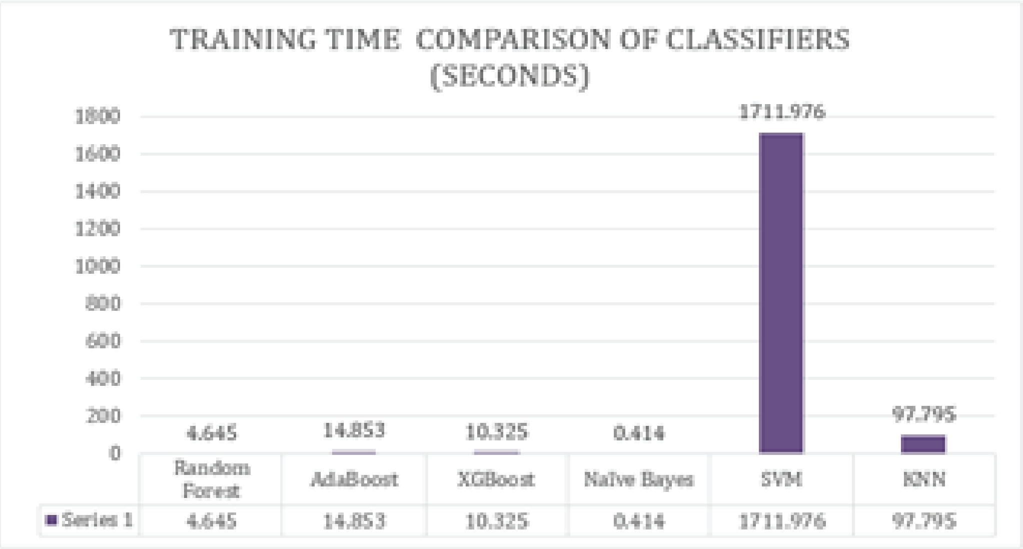


Figure 6: Execution Time Comparison of All Classifiers

The results from TABLE 1 highlight significant achievements, with AdaBoost and XGBoost showcasing exceptional performance at 100% accuracy, a notable milestone for DDoS attack detection. Following closely, KNN achieves a respectable accuracy of 99.93%, demonstrating its effectiveness. Random Forest secures the third position with 99.928% accuracy, establishing its capability for DDoS attack detection. Conversely, SVM and Naïve Bayes exhibit the lowest accuracies compared to other algorithms, registering 99.334% and 98.16%, respectively, indicating inadequate performance for DDoS attack detection.

The evaluation techniques for each algorithm used in this model are outlined in Table I. These techniques encompass accuracy as a percentage, F1-score, and training time measured in seconds.

Table 1: Evaluation Metrix

ALGORITHMS	EVALUATION		
	ACCURACY (percentage)	F1-SCORE	TRAINING TIME (seconds)
Random Forest	99.928	0.9928	4.645
AdaBoost	100	1	14.853
XGBoost	100	1	10.325
Naïve Bayes	98.16	0.6802	0.414
SVM	99.334	0.9288	1711.976
KNN	99.93	0.9927	97.795

The confusion matrix provides a concise summary of predicted outcomes in a classification scenario. It tabulates correct and incorrect predictions, assigning them to specific classes and quantifying their occurrences. This aids in the model training phase and subsequent evaluation of performance. A comparative analysis is conducted among different algorithms to detect DDoS attacks efficiently. Each classifier undergoes training and is evaluated, leading to the presentation of confusion matrices for each algorithm in this section. The overarching aim of the evaluation is to pinpoint the optimal classifier for the problem-solving model. Confusion matrices for each classifier are depicted in Figure 5 (A-F). Remarkably, all the considered ML algorithms, except Naïve Bayes, surpass our experimentation's expectations regarding accuracy, efficacy, and efficiency. It was observed that an imbalanced dataset skews Naïve Bayes' accuracy, causing it to classify more attacks than the count of harmless records predominantly. F1-score and recall metrics play a crucial role in comprehending false positive classifications.

Naïve Bayes' underperformance stems from its assumption of feature independence, grounded in Bayes' theorem, which contrasts with the dataset's actual feature interdependencies. Evaluation metrics, including F1-score, accuracy percentage, and training time in seconds for each algorithm utilized in this research experiment, are outlined in the Table I.

6. CONCLUSION AND FUTURE DIRECTIONS

This research introduces a robust DDoS detection model utilizing popular ML algorithms, including Random Forest, AdaBoost, XGBoost, Naïve Bayes, SVM, and KNN. The CICDDoS2019 dataset is categorized into "harmless" and "harmful" classes. All algorithms except Naïve Bayes effectively classify the dataset into these classes, displaying exceptional accuracy, efficiency, and training speed.

XGBoost and AdaBoost shine among these algorithms, exhibiting superior accuracy and F1 scores. Their training times show minor disparities. The proposed model achieves high accuracy and swiftness and identifies the top 20 influential features within the CICDDoS2019 dataset. By discarding irrelevant attributes, the research enhances accuracy, speed, and training efficiency. As future work, the model can be refined further by incorporating newer DDoS attack datasets and exploring hybrid mechanisms for improved detection capabilities. Building upon the current research, several promising avenues for further exploration and enhancement emerge:

- **Integration of Hybrid Approaches:** To elevate the model's detection capabilities, combining the strengths of multiple algorithms or integrating hybrid mechanisms could prove fruitful. Hybrid models that synergize the unique advantages of different algorithms may result in even more accurate and efficient DDoS attack detection.
- **Incorporation of Real-Time Analysis:** Expanding the model's applicability to real-time analysis can enhance its practical utility. The model could play a pivotal role in proactive threat mitigation by continuously monitoring network traffic and swiftly identifying potential DDoS attacks in real time.
- **Leveraging Advanced Feature Engineering:** Exploring advanced feature engineering techniques can further refine the model's predic-

tive power. Incorporating domain-specific knowledge and extracting more relevant features may contribute to more nuanced and accurate predictions.

• **Utilization of Deep Learning Techniques:** Integrating deep learning methodologies, such as convolutional neural networks (CNNs) or recurrent neural networks (RNNs), can provide the model with enhanced capabilities to capture complex patterns and relationships within network data, potentially leading to improved accuracy.

• **Evaluation with Diverse Datasets:** Testing the model against a wider range of diverse datasets containing various types of DDoS attacks and network scenarios can validate its robustness and generalizability. Incorporating data from different sources and attack scenarios can ensure its effectiveness across different contexts.

• **Optimization for Scalability:** Optimizing the model's scalability becomes crucial as network infrastructures continue to grow in complexity and scale. Developing strategies to handle large-scale networks and big data environments while maintaining accuracy and efficiency will be essential.

• **Continuous Model Enhancement:** The model should undergo ongoing refinement and updates to stay relevant in the face of evolving attack strategies. Regularly updating the model with new attack patterns, techniques, and datasets ensures its effectiveness against emerging threats.

• **Collaboration and Knowledge Sharing:** Collaborating with other researchers, practitioners, and industry experts can lead to innovative insights and solutions. Sharing knowledge and experiences can drive the advancement of DDoS detection techniques.

These future directions can further elevate the DDoS detection model's capabilities, making it a more potent tool in safeguarding networks against the evolving landscape of cyber threats.

REFERENCES

[1] M. A. Khan and K. Salah, "IoT security: Review, blockchain solutions, and open challenges," *Futur. Gener. Comput. Syst.,* vol. 82, pp. 395–411, doi: 10.1016/j.future.2017.11.022, 2018.

[2] M. Al-Sarem et al., "An aggregated mutual information based feature selection with machine learning methods for enhancing iot botnet attack detection," *Sensors,* vol. 22, no. 1, doi: 10.3390/s22010185, 2022.

[3] J. Li et al., "RTED-SD: A Real-Time Edge Detection Scheme for Sybil DDoS in the Internet of Vehicles," *IEEE Access,* vol. 9, pp.11296-11305, doi: 10.1109/ACCESS.2021.3049830, 2021.

[4] A. Aljuhani, "Machine Learning Approaches for Combating Distributed Denial of Service Attacks in Modern Networking Environments," *IEEE Access,* vol. 9, pp. 42236–42264, doi: 10.1109/ACCESS.2021.3062909, 2021.

[5] T. Mahjabin et al., "A survey of distributed denial-of-service attack, prevention, and mitigation techniques," *Int. J. Distrib. Sens. Networks,* vol. 13, no. 12, doi: 10.1177/1550147717741463, 2017.

[6] E. SÖĞÜT et al., "Farklı Türde Dağıtık Hizmet Dışı Bırakma Saldırılarının Tespiti," *Gazi Üniversitesi Fen Bilim. Derg. Part C Tasarım ve Teknol.,* vol.9, no.1, doi: 10.29109/gujsc.840126, March, 2021.

[7] S. N. Thanh et al., "Survey on botnets: Incentives, evolution, detection and current trends," *Futur. Internet,* vol. 13, no. 8, doi: 10.3390/fi13080198, 2021.

[8] M. A. Khan et al., "HCRNNIDS: Hybrid Convolutional Recurrent Neural," *Multidiscip Digit. Publ Inst.,* 2021.

[9] S. Chen et al., "A vision of IoT: Applications, challenges, and opportunities with China Perspective," *IEEE Internet Things J.,* vol. 1, no. 4, pp. 349–359, doi: 10.1109/JIOT.2014.2337336, 2014.

[10] R. Santos et al., "Machine learning algorithms to detect DDoS attacks in SDN," *Concurr. Comput. Pract. Exp.,* vol. 32, no. 16, pp. 1–14, doi: 10.1002/cpe.5402, 2020.

[11] A. E. Bouaouad et al., "The key layers of IoT architecture," Proc. 2020 *5th Int. Conf. Cloud Comput. Artif. Intell. Technol. Appl. CloudTech 2020,* pp. 20–23, doi: 10.1109/CloudTech49835.2020.9365919, 2020.

[12] P. Lou et al., “An anomaly detection method for cloud service platform,” *ACM Int. Conf. Proceeding Ser.,* pp. 70–75, doi: 10.1145/3340997.3341005, 2019.

[13] M. A. Underwood et al., “nternet of things: Toward smart networked systems and societies,” *Appl. Ontol.,* vol. 10, no. 3–4, pp. 355–365, doi: 10.3233/AO-150153, 2015.

[14] K. Sha et al., “A survey of edge computing-based designs for IoT security,” *Digit. Commun. Networks,* vol. 6, no. 2, pp. 195–202, doi: 10.1016/j.dcan.2019.08.006, 2020.

[15] C. L. Zhong et al., “Study on the IOT architecture and gateway technology,” *Proc. - 14th Int. Symp. Distrib. Comput. Appl. Business, Eng. Sci. DCABES 2015,* pp. 196–199, doi: 10.1109/DCABES.2015.56, 2016.

[16] I. Grønbæk, “Architecture for the Internet of Things (IoT): API and interconnect,” *Proc. - 2nd Int. Conf. Sens. Technol. Appl., SENSORCOMM 2008, Incl. MESH 2008 Conf. Mesh Networks; ENOPT 2008 Energy Optim. Wirel. Sensors Networks, UNWAT 2008 Under Water Sensors Syst.,* pp. 802–807, doi: 10.1109/SENSORCOMM.2008.20, 2008.

[17] Soma Bandyopadhyay et al., “Role Of Middleware For Internet Of Things: A Study,” *Int. J. Comput. Sci. Eng. Surv.,* vol. 2, no. 3, pp. 94–105, doi: 10.5121/ijcses.2011.2307, 2011.

[18] V. Jakkula, “Tutorial on Support Vector Machine (SVM),” *Sch. EECS, Washingt. State Univ.,* pp. 1–13, *[Online],* Available: http://www.ccs.neu.edu/course/cs5100f11/resources/jakkula.pdf, 2011.

[19] S. Dong and M. Sarem, “DDoS Attack Detection Method Based on Improved KNN with the Degree of DDoS Attack in Software-Defined Networks,” *IEEE Access,* vol. 8, pp. 5039–5048, doi: 10.1109/ACCESS.2019.2963077, 2020.

[20]. J. Xue et al., “Bound maxima as a traffic feature under DDOS flood attacks,” *Math. Probl. Eng.,* vol. 2012, no.1, pp. 1–5, doi: 10.1155/2012/419319, 2012.

[21] S. Wankhede and D. Kshirsagar, “DoS Attack Detection Using Machine Learning and Neural Network,” *Proc. - 2018 4th Int. Conf. Comput. Commun. Control Autom. ICCUBEA 2018, pp. 1-5* doi: 10.1109/ICCUBEA.2018.8697702, 2018.

[22] M. A. Prriyadarshini and S. R. Devi, “Detection of DDoS attacks using supervised learning technique,” *J. Phys. Conf. Ser.,* vol. 1716, no. 1, doi: 10.1088/1742-6596/1716/1/012057, 2021.

[23] N. V. Patil et al., “E-Had: A distributed and collaborative detection framework for early detection of DDoS attacks,” *J. King Saud Univ. - Comput. Inf. Sci.,* doi: 10.1016/j.jksuci.2019.06.016, 2019.

[24] R. Bose, “Virtual labs project: A paradigm shift in internet-based remote experimentation,” *IEEE Access,* vol. 1, pp. 718–725, doi: 10.1109/ACCESS.2013.2286202, 2013.

[25] R. Doshi et al., “Machine learning DDoS detection for consumer internet of things devices,” *Proc. - 2018 IEEE Symp. Secur. Priv. Work. SPW 2018, no. Ml*, pp. 29–35, doi: 10.1109/SPW.2018.00013, 2018.

[26] M. Sajawal et al., “A Predictive Analysis of Retail Sales Forecasting using Machine Learning Techniques,” L*ahore Garrison University Research Journal of Computer Science and Information Technology,* vol. 6, no. 4, pp. 33-45, https://doi.org/10.54692/lgurjcsit.2022.0604399, 2022.

[27] M. U. Ashraf et al., “Comparative Analysis of Machine Learning Techniques for Predicting Air Pollution,” L*ahore Garrison University Research Journal of Computer Science and Information Technology,* vol. 6, no. 2, pp. 40-54, https://doi.org/10.54692/lgurjcsit.2022.0602270, 2022.

[28] M. Ahmed et al., “A Machine Learning-Based Tool for Performance Optimization of Parallel SPMV Computations Using Block CSR,” *Appl. Sci.*, vol. 12, no. 14, pp. 7073. https://-doi.org/10.3390/app12147073, 2022.

[29] S. Usman et al., “ZAKI: A Smart Method and Tool for Automatic Performance Optimization of Parallel SpMV Computations on

Distributed Memory Machines," *Mobile Netw Appl (2019).* https://-doi.org/10.1007/s11036-019-01318-3, 2023.

[30] S. Usman et al., "ZAKI+: A Machine Learning Based Process Mapping Tool for SpMV Computations on Distributed Memory Architectures," *in IEEE Access,* vol. 7, pp. 81279-81296, doi: 10.1109/ACCESS.2019.2923565, 2019.

[31] M. U. Ashraf, "A Survey on Data Security in Cloud Computing Using Blockchain: Challenges, Existing-State-Of-The-Art Methods, And Future Directions," *Lahore Garrison University Research Journal of Computer Science and Information Technology,* vol. 5, no. 3, pp. 15-30, 2021.

[32] M. U. Ashraf et al., "A Survey on Emotion Detection from Text in Social Media Platforms," *Lahore Garrison University Research Journal of Computer Science and Information Technology,* vol. 5, no. 2, pp. 48-61, 21 Jun 2021.

[33] K. Shinan et al., "Machine learning-based botnet detection in software-defined network: a systematic review," *Symmetry* vol. 13, no. 5, pp. 866, 2021.

[34] A. Hannan et al., "A decentralized hybrid computing consumer authentication framework for a reliable drone delivery as a service," *Plos one,* vol. 16, no, 4, pp. e0250737, 2021.

[35] S. Fayyaz et al., "Solution of combined economic emission dispatch problem using improved and chaotic population-based polar bear optimization algorithm," *IEEE Access,* vol. 9, pp. 56152-56167, 2021.

[36] I. Hirra et al., "Breast cancer classification from histopathological images using patch-based deep learning modeling," *IEEE Access*, vol. 9, pp. 24273-87, 2 Feb 2021.

[37] M. U. Ashraf et al., "AAP4All: An Adaptive Auto Parallelization of Serial Code for HPC Systems," *INTELLIGENT AUTOMATION AND SOFT COMPUTING,* vol. 30, no. 2, pp.615-39, 1 Jan 2021.

[38] T. Hafeez et al., "EEG in game user analysis: A framework for expertise classification during gameplay," *Plos one,* vol. 16, no. 6, pp. e0246913, 18 Jun 2021.

[39] N. Siddiqui et al., "A highly nonlinear substitution-box (S-box) design using action of modular group on a projective line over a finite field," *Plos one*, vol. 15, no.11,pp. e0241890, 12 Nov 2020 .

[40] M. U. Ashraf et al., "Detection and tracking contagion using IoT-edge technologies: Confronting COVID-19 pandemic," *2020 international conference on electrical, communication, and computer engineering (ICECCE),* pp. 1-6, IEEE, 2020.

[41] K. Alsubhi et al., "MEACC: an energy-efficient framework for smart devices using cloud computing systems," *Frontiers of Information Technology & Electronic Engineering,* vol. 21, no. 6, pp. 917-930, 2020.

[42] S. Riaz et al., "A Comparative Study of Big Data Tools and Deployment PIatforms.," *In 2020 International Conference on Engineering and Emerging Technologies (ICEET),* pp. 1-6, IEEE, 22 Feb 2020.

[43] M. U. Ashraf et al., "Empirical investigation: performance and power-consumption based dual-level model for exascale computing systems," *IET Software*, vol.14, no. 4, pp. 319-27, 27 Jul 2020.

[44] M. U. Ashraf et al., "IDP: A Privacy Provisioning Framework for TIP Attributes in Trusted Third Party-based Location-based Services Systems," *International Journal of Advanced Computer Science and Applications (IJACSA),* vol. 11, no. 7, pp. 604-617, 2020.

[45] A. Manzoor et al., "Inferring Emotion Tags from Object Images Using Convolutional Neural Network," *Applied Sciences,* vol. 10, no.15, pp. 5333, 2020.

[46] K. Alsubhi et al., "A Tool for Translating sequential source code to parallel code written in C++ and OpenACC," *In 2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA)*, pp. 1-8, IEEE, 2019.

[47] M. U. Ashraf et al., "H2E: A Privacy

Provisioning Framework for Collaborative Filtering Recommender System," *International Journal of Modern Education and Computer Science*, vol.11, no. 9, pp. 1,1 Sep 2019.

[48] M. U. Ashraf et al., "A Roadmap: Towards Security Challenges, Prevention Mechanisms for Fog Computing," *In 2019 International Conference on Electrical, Communication, and Computer Engineering (ICECCE),* pp. 1-9, IEEE, 24 Jul 2019.

[49] M. U. Ashraf et al., "State-of-the-art Challenges: Privacy Provisioning in TPP Location Based Services Systems," *International Journal of Advanced Research in Computer Science (IJARCS)*, vol. 10, no. 2, pp. 68-75, 20 Apr 2019.

[50]. M. U. Ashraf et al., "Improving Performance In Hpc System Under Power Consumptions Limitations," *International Journal of Advanced Research in Computer Science*, vol. 10, no. 2, Mar 2019.

[51] R. Javed et al., "Prediction and monitoring agents using weblogs for improved disaster recovery in cloud," *Int. J. Inf. Technol. Comput. Sci.(IJITCS),* vol.11, no. 4, pp. 9-17, 2019.

[52] M. Ali et al., "Prediction of Churning Behavior of Customers in Telecom Sector Using Supervised Learning Techniques," *In 2018 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)*, pp. 1-6, IEEE, 2018.

[53] M. U. Ashraf et al., "Performance and power efficient massive parallel computational model for HPC heterogeneous exascale systems," *IEEE Access*, 6, pp. 23095-107, 9 Apr 2018.

[54] M. U. Ashraf et al., "Toward exascale computing systems: An energy efficient massive parallel computational model," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 2.

[55] M. U. Ashraf., "Provisioning quality of service for multimedia applications in cloud computing," *Int. J. Inf. Technol. Comput. Sci.(IJITCS)*, vol. 10, no. 5, pp.40-7, 2018.

[56] M. U. Ashraf et al., "Efficient Execution of Smart City's Assets Through a Massive Parallel Computational Model," *InInternational Conference on Smart Cities, Infrastructure, Technologies and Applications,* pp. 44-51, *Springer,* Cham, 27 Nov 2017.

[57] M. S. Alrahhal et al., "AES-route server model for location based services in road networks," *International Journal Of Advanced Computer Science And Applications,* vol. 8, no. 8, pp. 361-368, 2017.

[58] M. U. Ashraf et al., "High performance 2-D Laplace equation solver through massive hybrid parallelism," *In 2017 8th International Conference on Information Technology (ICIT),* pp. 594-598, IEEE, 17 May 2017.

[59] M. Mumtaz et al., "Iteration Causes, Impact, and Timing in Software Development Lifecycle: SLR," *IEEE Access,* vol. 10, pp. 65355-65375, 2022.

[60] M. Ahmad et al., "Efficient Liver Segmentation from Computed Tomography Images Using Deep Learning," *Computational Intelligence and Neuroscience 2022,* no. 1, pp.2665283, 2022.

[61] H. Tufail et al., "The Effect of Fake Reviews on e-Commerce During and After Covid-19 Pandemic: SKL-Based Fake Reviews Detection," *IEEE Access,* vol. 10, pp. 25555-25564, 2022.