IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

REAL-TIME DDOS ATTACK DETECTION SYSTEM USING APACHE FLINK AND GRADINET BOOSTING ALGORITHM

Rahulkumar Choudhary[1], Kartik Shikhare [2], Akanksha Mate [3], Sanket Gawali [4] , Rupal More [5]

Abstract: DDoS attacks involve routing large volumes of traffic from multiple computers to a network or server with the intention of crashing the system or disrupting its function. The attacks overwhelm the devices, services, and network of the target with fake internet traffic, rendering them inaccessible or useless to legitimate users. These attacks mimic a flood of requests made by browsers to load a web page. The impact of DDoS attacks can be severe, leading to downtime and loss of revenue for businesses. Early detection of DDoS attacks is crucial as it helps protect the functioning and security of a network. Networks without a robust DDoS defence strategy may struggle to defend against the range of attacks, which can be challenging to trace. Addressing the problem requires models that can manage the time information in network traffic flows. By detecting attacks during the initial requests to the server, we can block requests made by suspicious IP addresses. If they are false DDoS attack warnings, we can scale up our application to handle the traffic. To achieve real-time detection and better attack classification, we propose using Apache Flink and Gradient Boosting, respectively. We have achieved a mean accuracy of 99.41% of the model with big data approach and other enhancements. The minimum average training and testing time in minutes was 14.08 and 0.036, respectively. We can detect an attack in real-time in few milliseconds.

Keywords: Distributed Denial of Service attack detection; machine learning; network security.

I. Introduction

DDoS attacks are a type of cyber-attack in which a large volume of traffic from multiple sources is directed towards a target network or server with the goal of overwhelming it and causing it to crash. These attacks are a significant threat to businesses, as they can render websites and web applications inaccessible to legitimate users, leading to significant revenue losses.

The technique used in DDoS attacks involves flooding the target with fake internet traffic that looks like legitimate requests from browsers. The aim is to overwhelm the devices, services, and network of the target, thereby rendering them useless for their intended purpose.

Early detection of DDoS attacks is critical for businesses as it allows them to take timely measures to protect their network's functioning and security. However, networks without robust DDoS defence strategies may struggle to defend against the wide range of DDoS attacks, which can be challenging to trace. Therefore, it is essential to develop models that can manage the time information contained in network traffic flows to address these problems.

One effective approach is to detect the attack while the initial requests are being made to the server and block requests from suspicious IP addresses. In case of false DDoS attack warnings, it is also possible to scale up the application to handle the increased traffic.

To achieve real-time detection and better attack classification, the proposed system will use Apache Flink for real-time detection and Gradient Boosting for improved attack classification. It is crucial to have a system that operates in real-time to prevent system crashes, and thus the proposed system will aim to be as real-time as possible.

We have used ACK/PUSH-ACK DDoS dataset to train model which contains around 0.5 million rows and 27 columns to create and accurate model which predicts output in Realtime.

II. Related Work

Nishanth, N. [1] Wireless ad hoc networks are vulnerable to various attacks due to lack of centralized management and absence of secure boundaries. The flooding-based denial-of-service (DoS) attack targets the mobile nodes' limited resources, resulting in excess battery consumption. SYN flooding-based DoS attack overflows target buffers and creates network congestion by sending many spoofed SYN packets. In this article, we propose a method for detecting SYN flooding attacks in wireless ad hoc networks. The method involves mathematical modeling using Bayesian inference, proving its equivalence with exponential weighted moving average, and developing an efficient algorithm. The proposed method provides higher detection accuracy and extremely lower false detection rates in defending against flooding-based DoS attacks.

Polat, H. [2] SDN has security issues when facing DDoS attacks, as it can overload the controller, leading to network performance degradation. This study proposes machine learning-based models for DDoS attack detection in SDN. Features were extracted from normal and attack traffic datasets, and a new dataset was created using feature selection methods. SVM, NB, ANN, and KNN classifiers were used to train and test both datasets. The wrapper feature selection with a KNN classifier achieved the highest accuracy rate (98.3%) in DDoS attack detection. Machine learning and feature selection algorithms can improve DDoS attack detection in SDN, reducing processing loads and times.

Ganguly, S. [3] Proposing new algorithms for real-time detection of DDoS attacks in large ISP networks using hash-based data structures to efficiently track destination IP addresses. Our solution addresses the challenge of efficiently tracking top distinct-source frequencies over a stream of updates and distinguishing between DDoS activity and flash crowds. Results show the effectiveness of our approach.

Ahmad et al. [7] evaluated various machine learning models for detecting intrusions in the network system. Authors evaluated the SVM, RF, and Extreme Learning Machine (ELM) models for accuracy, recall, and precision. The train test split was used for the training and testing purpose at 80% for training and 20% for testing. The results showed that ELM outperforms other approaches with higher accuracy and precision of 95.5% on full sample data, and recall is slightly lower compared with SVM and RF.

Kato and Klyuev [8] studied the major problems of network intrusion detection using machine learning algorithms. The authors focused on designing a practically intelligent intrusion detection system having defined accuracy and a low false-positive rate. The results showed that the proposed approach could develop the intelligent IDS and achieved an accuracy of 86.2%, a false-positive rate of 13%, and a true negative rate of 87%.

Syed et al. [9] applied an ML-based application layer DoS attack detection framework for the detection of DoS attack on Message Queuing Telemetry Transport (MQTT) data communication protocol. The ML models applied for the detection of attack were Decision Trees (C4.5), Multi-Layer Perceptron (MLP) and Average One-Dependence Estimator (AODE). The results showed that the C45 outperforms other with an accuracy of 99.09% than the AODE and MLP with accuracy of 99.00% and 95.93%, respectively, using limited features. The recall of C45 achieves a high score of 99.1% with respect to AODE and MLP, with 99% and 95.9%, respectively.

Priya et al. [10] proposed an ML-based model for the detection of DDoS attacks. The authors applied three different machine learning models: K-Nearest Neighbors (KNN), Random Forest (RF) and Naive Bayes (NB) classifier. The proposed approach can detect any type of DDoS attack in the network. The results of the proposed approach showed that the model can detect attacks with an average accuracy of 98.5%.

Ujjan et al. [11] proposed entropy-based DoS detection to identify features of traffic DoS by combining two entropies through Stacked auto encoder (SAE) and CNN. The CPU utilization was much higher and time-consuming. The accuracies of the models were 94% and 93%, respectively.

Gadze et al. [12] proposed deep learning models to detect and mitigate the risk of DDoS attacks that target the centralized controller in Software Defined Network (SDN) through Long Short-Term Memory (LSTM) and Convolutional Neural Network (CNN). The accuracies of the models were lower. LSTM and CNN were 89.63% and 66%, respectively, when data spitted was in 70/30 ratio. However, in the case of LSTM model to detect TCP, UDP and ICMP, DDoS was the most time-consuming among the 10 attempts.

Ahuja et al. [13] proposed a hybrid ML model Support Vector Classifier with Random Forest (SVC-RF) for the classification of traffic as BENIGN or DDoS. The authors extracted the number of features from the original dataset and created a new dataset called SDN dataset with novel features. The proposed model results showed that the classifier SVC-RF can successfully classify the traffic with an accuracy of 98.8% using SDN dataset.

Dehkordi et al. [14] proposed a method to detect DDoS attack in Software Defined Network (SDN) using different machine learning models. The proposed method contained three main sections: (1) collect; (2) entropy-based; and (3) classification. The proposed method was applied on three different datasets. The results showed that by applying the ML models Logistic algorithms, J48 algorithm, BayesNet algorithm, Random Tree algorithm and REPTree algorithm, the average accuracy achieved was 99.62% 99.87%, 99.33%, 99.8% and 99.88%, respectively by application on an ISCX-SlowDDos2016 dataset.

Yifei Chen [20] evaluated the performance of GBMCI against other popular survival models using a large-scale breast cancer prognosis dataset. Our experiment showed that GBMCI consistently outperformed other methods across various covariate settings.

Syed, N.F. [30] IoT relies on effective data communication protocols, such as the popular MQTT protocol. However, MQTT is vulnerable to Application layer Denial of Service (DoS) attacks, which can cause widespread disruption. In this paper, a machine learning-based detection framework for detecting protocol-based Application layer DoS attacks against MQTT message brokers is proposed. The framework was tested on legitimate and protocol-compliant DoS attack scenarios, and the results showed that attackers can overwhelm server resources even with legitimate access being denied. Features identified in MQTT were shown to have high attack detection accuracy, with field size and length-based features reducing false-positive rates and being suitable for detecting IoT-based attacks.

III. Material and Methodology

A. Dataset

The source of our data was the ACK/PUSH-ACK DDoS dataset, which is publicly available on Kaggle. This dataset falls into the category of large datasets and contains approximately 0.5 million records with 27 feature columns and a target column. It includes three main labels: DDoS-PSH-ACK, DDoS-ACK, and Benign [22].

B. Our Approach and Data Pre-Processing

The block diagram in Figure 1 illustrates our system, which employs classification algorithms following pre-processing to detect DDoS attacks. It comprises four primary components: data collection, data pre-processing, machine learning model, and an evaluation process that generates the output indicating whether the attack is a DDoS attack.

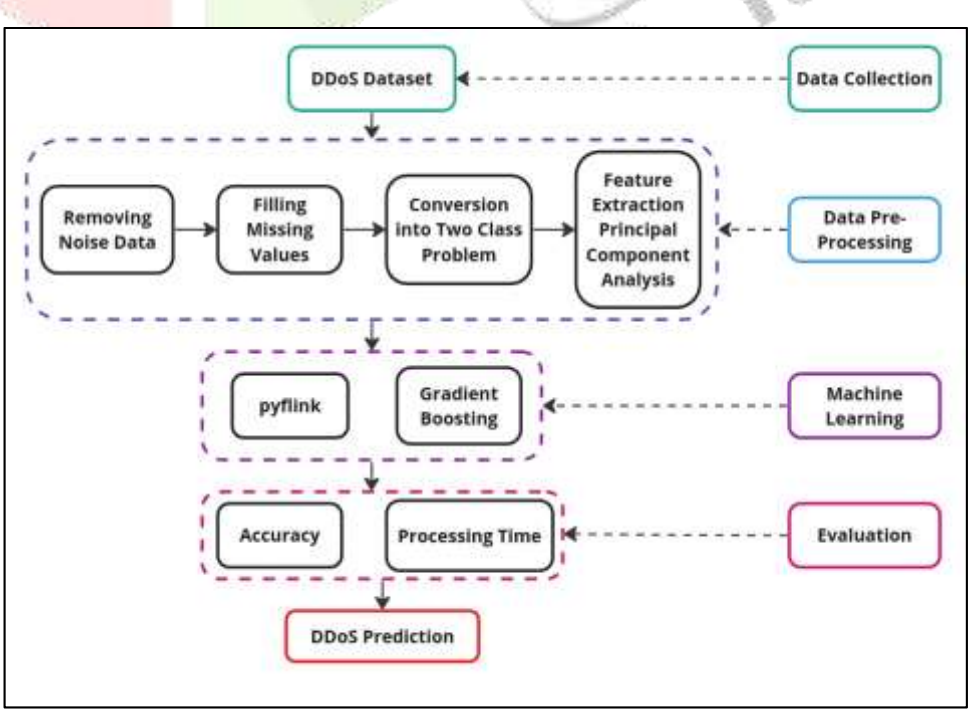

Figure 1. System block diagram to predict DDoS Attacks.

Our proposed model comprises four components, the first of which is data collection. In this component, we obtained the dataset to be used in the pre-processing phase. The second component, pre-processing, involved several procedures to prepare the data for machine learning. We began with noise filtering, which aims to reduce noise in the data collected during development. Next, we handled missing values using various strategies, including ignoring data with missing entries or replacing them with a consistent method. The third step involved transforming the 3-class problem into a 2-class problem. Specifically, we grouped "DDoS-PSH-ACK" and "DDoS-ACK" as one class and "Benign" as the other class.

In the fourth step, we used Principal Component Analysis (PCA) to reduce the number of features from 27 to 9, selecting the most appropriate features for predicting the model. We also converted all string type data to float and integer type. We used the pyflink library to create a Flink execution environment and a table environment, defined the CSV file schema, and created a CSV table source using the csv table source builder. We registered the CSV table source as a table in the table environment using the connect method and the with format and with schema descriptors. We then queried the table using the from path method and selected the columns we wanted to load into the model. For training and testing the data, we employed the Gradient Boosting algorithm as the machine learning model. Finally, we evaluated the approach using various metrics, including accuracy, to measure the model's performance.

C. Classification Machine Learning Models

Our approach to building a machine learning model involved using the Gradient Boosting algorithm and a big data approach.

(a) Gradient Boosting

Gradient boosting is a widely used ensemble learning method that is highly effective in a variety of fields. The basic premise behind gradient boosting is to iteratively build a set of "weak" learners, typically decision trees, that are trained to correct the errors of the previous learners. This process continues until a pre-specified stopping criterion is met, such as a maximum number of learners or a desired level of performance. One of the key advantages of gradient boosting is its ability to handle complex, non-linear relationships between features and the target variable. Additionally, gradient boosting can be used for both regression and classification tasks. During the learning phase, gradient boosting involves fitting a sequence of decision trees to the training data. At each iteration, the algorithm calculates the gradient of the loss function with respect to the current model's predictions and then fits a new decision tree to the negative gradient. The resulting model is a weighted combination of the previous models, with each tree assigned a weight proportional to its contribution to the overall prediction. Like other ensemble methods, gradient boosting can also suffer from overfitting if the models are too complex, or the training data is noisy. Regularization techniques, such as limiting the depth of the decision trees or adding penalties to the loss function, can be used to address this issue.

Equation for gradient boosting is as follows,

$$F_m = \mathrm{F}_{m-1}(x) + \gamma_m h_m(x)$$

Gradient boosting helps us in predicting the model in faster time frame as compared to random forest and other such boosting methods. Gradient boosting works in steps of preoptimizing steps; the current step enhances the output generated from previous step in such way the gradient boosting predicts output in much shorter time span.

We utilized the GradientBoostingClassifier from the Python library sklearn to train our model. This classifier has several tunable parameters, including n_estimators, learning_rate, and max_depth. In our training process, we set these parameters to 100, 0.2, and 4, respectively. By using these values, we were able to obtain model predictions that were optimized for our dataset and environmental needs. It's worth noting that the learning_rate and max_depth parameters can be adjusted to different values based on the size and complexity of the dataset, as well as the specific requirements of the environment in which the model will be used.

IV. Experiments and Results

A. *Experiment Setup*

In our experimental setup, we utilized various attackers that were controlled from the central subsystem of the complete system. The primary system consists of a frontend comprising only two routes: an attacker route and a prediction route that displays the output of the current predictions.

To initiate the attack, we started the attacker machines that were distributed across different configurations. A REST API service was created using FastAPI, which helped in initiating the attack with the help of endpoints. To connect it to the central system, we used port forwarding with a public IP, namely ngrok. To create an attacker machine and launch an attack, we added the public link of the setup into the GUI. In the background of this setup, when a DDoS attack is initiated to the requested resource, a service named tshark records the incoming calls to the system and stores it in CSV format. After a certain number of requests, the model picks up this CSV to predict the normal calls and DDoS attack calls. Finally, the prediction is published to the web app with the help of a Realtime database provided by Firebase.

B. *Evaluation Results*

We evaluated the results of our experiment based on several metrics, including accuracy, precision, F1 score, and confusion matrix. The accuracy of the Gradient Boosting Classifier is illustrated in Figure 2.

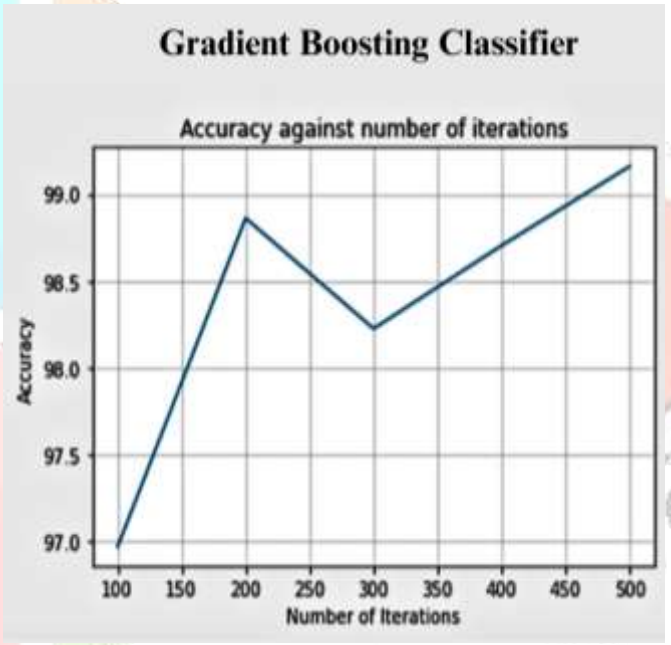

Figure 2. Evaluation Results of Gradient Boosting Classifier

We selected the Gradient Boosting classifier as our model.

Evaluation Metrics	Gradient Boosting Classifier
Accuracy	99.41%
Precision	99.41%
F1 Score	99.39%
False-Positive Rate	0.04%
False-Negative Rate	0.02%

Table 1. Evaluation matrix.

During the training phase, we experimented with different numbers of trees, ranging from 100 to 400. We observed that the accuracy of the model was at its lowest, approximately 99.7%, when trained with 100 trees, while the highest accuracy of 99.98% was achieved when the model was trained with 400 trees.

C. *Execution Time*

The execution time of the Gradient Boosting classifier (GB) is illustrated in Figure 3. During our experiments, we observed that the minimum testing time of approximately 0.2 minutes was achieved when the GB model was trained

with 100 trees, while the maximum testing time of around 0.8 minutes was recorded when the model was trained with 400 trees.

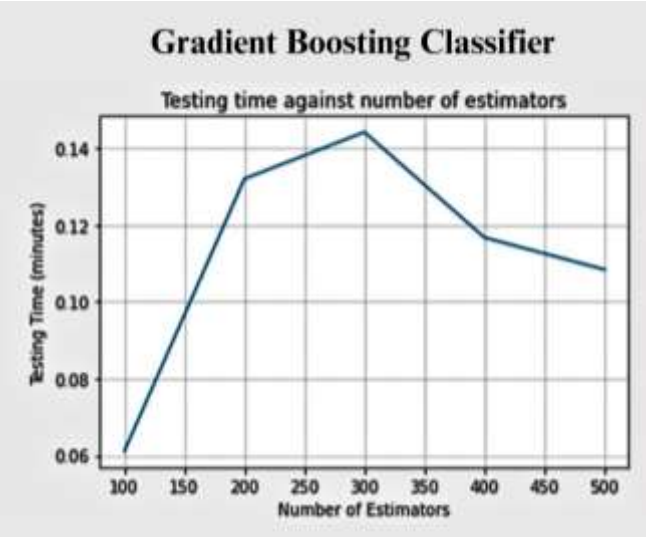

Figure 3. Execution time of Gradient Boosting Classifier

V. Discussion

Table 2 shows compilation of recent studies related to DDOS that have employed various machine learning, deep learning, and big data approaches. The table provides details of the accuracies and execution time achieved in each of these studies.

Studies and Year	Model	Accuracy%	Execution Time (s)	Big Data Framework
Saravanan, 2020	Logistic Regression	93.90%	0.48	
Zhang, Dai, Li and Zhang [29], 2018	Random Forest	97.4%	1.10	Spark, (IDS detection)
Wang, Xiao and Long, 2017 [26]	PCA-SVM	86.31%	-	Spark, (IDS detection)
Halimaa and Sundarakantham , 2019	SVM	93.95%	-	-
Dehkordi, Soltanaghaei and Boroujeni,[14] 2021	Logistic algorithms	99.62%	1.19	-
Priya, Sivaram, Yuvaraj and Jayanthiladevi , 2020[10]	Naive Bayes	98.50%	-	-
Mazhar Javed Awan,2021	Random Forest	99.34%	0.11	Spark Ml
Mazhar Javed Awan,2021	Multi-Layer Perceptron	99.38%	0.04	Spark Ml
Our Approach	Gradient Boosting	99.41%	0.036	Flink

Table 2. State-of-the-art comparison work with our approaches in terms of accuracy and execution time.

Among the studies listed, the logistic regression model used in Saravanan's study in 2020 achieved an accuracy of 93.9% with an execution time of 0.48 seconds, whereas the random forest model used by Zhang et al. in 2018 achieved an accuracy of 97.4% with an execution time of 1.10 seconds on Spark big data framework for intrusion detection system. Another study by Halimaa and Sundarakantham in 2019 achieved an accuracy of 93.95% using Support Vector Machine (SVM) model, but the execution time and big data framework used were not mentioned. Dehkordi, Soltanaghaei, and Boroujeni in 2021 achieved a higher accuracy of 99.62% using logistic algorithms, but their study did not mention the big data framework used. Priya, Sivaram, Yuvaraj, and Jayanthiladevi in 2020 achieved an accuracy of 98.5% using Naive Bayes model, but the execution time and big data framework used were not reported. Mazhar Javed Awan conducted two studies in 2021 using Spark ML framework, where they

achieved an accuracy of 99.94% and 99.38% using Random Forest and Multi-Layer Perceptron models, respectively. Finally, our approach study used Gradient Boosting model on Flink big data framework and achieved an accuracy of 99.41% with an execution time of 0.036 seconds.

VI. Conclusion

Traditional intrusion detection techniques may work well on slow-speed or small data, but they are inefficient in handling high-speed or large data. Therefore, new approaches capable of detecting intrusion signs in large data are necessary. This paper presents a real-time machine learning model for predicting DDoS attacks using Apache Flink and the Gradient Boosting classification algorithm approach. To enhance the model's performance, we utilized the distributed system Apache Flink through the pyflink library and the Gradient Boosting classification algorithm. In addition to the detection of DDoS attacks, using Apache Flink and Gradient Boosting algorithm we have optimized the performance of the models by minimizing the prediction time as compared with other existing approaches.

In the future, we plan to train model with deep learning approaches such as neural networks to predict real-time results from convolutional neural network architectures.

References

1. Nishanth, N.; Mujeeb, A. Modeling and detection of flooding-based denial-of-service attack in wireless ad hoc network using Bayesian inference. IEEE Syst. J. 2020, 15, 17–26.3.
2. Polat, H.; Polat, O.; Cetin, A. Detecting DDoS Attacks in Software-Defined Networks Through Feature Selection Methods and Machine Learning Models. Sustainability 2020, 12, 1035.
3. Ganguly, S.; Garofalakis, M.; Rastogi, R.; Sabnani, K. Streaming algorithms for robust, real-time detection of ddos attacks. In Proceedings of the 27th International Conference on Distributed Computing Systems (ICDCS'07), Toronto, ON, Canada, 25–27 June 2007; p. 4.
4. Awan, M.J.; Rahim, M.S.M.; Nobanee, H.; Munawar, A.; Yasin, A.; Azlanmz, A.M.Z. social media and Stock Market Prediction: A Big Data Approach. Comput. Mater. Contin. 2021, 67, 2569–2583.
5. Awan, M.J.; Gilani, S.A.H.; Ramzan, H.; Nobanee, H.; Yasin, A.; Zain, A.M.; Javed, R. Cricket Match Analytics Using the Big Data Approach. Electronics 2021, 10, 2350.
6. Khalil, A.; Awan, M.J.; Yasin, A.; Singh, V.P.; Shehzad, H.M.F. Flight Web Searches Analytics through Big Data. Int. J. Comput. Appl. Technol. 2021, in press.
7. Ahmad, I.; Basheri, M.; Iqbal, M.J.; Rahim, A. Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection. IEEE Access 2018, 6, 33789–33795.
8. Kato, K.; Klyuev, V. Development of a network intrusion detection system using Apache Hadoop and Spark. In Proceedings of the 2017 IEEE Conference on Dependable and Secure Computing, Taipei, Taiwan, 7–10 August 2017; pp. 416–423.
9. Syed, N.F.; Baig, Z.; Ibrahim, A.; Valli, C. Denial of service attack detection through machine learning for the IoT. J. Inf. Telecommun. 2020, 4, 482–503.
10. Priya, S.S.; Sivaram, M.; Yuvaraj, D.; Jayanthiladevi, A. Machine learning based DDoS detection. In Proceedings of the 2020 International Conference on Emerging Smart Computing and Informatics (ESCI), Pune, India, 12–14 March 2020; pp. 234–237.
11. Ujjan, R.M.A.; Pervez, Z.; Dahal, K.; Khan, W.A.; Khattak, A.M.; Hayat, B. Entropy Based Features Distribution for Anti-DDoS Model in SDN. Sustainability 2021, 13, 1522.
12. Gadze, J.D.; Bamfo-Asante, A.A.; Agyemang, J.O.; Nunoo-Mensah, H.; Opare, K.A.-B. An Investigation into the Application of Deep Learning in the Detection and Mitigation of DDOS Attack on SDN Controllers. Technologies 2021, 9, 14.
13. Ahuja, N.; Singal, G.; Mukhopadhyay, D.; Kumar, N. Automated DDOS attack detection in software defined networking. J. Netw. Comput. Appl. 2021, 187, 103108.
14. Dehkordi, A.B.; Soltanaghaei, M.; Boroujeni, F.Z. The DDoS attacks detection through machine learning and statistical methods in SDN. J. Supercomput. 2021, 77, 2383–2415.

15. Elham Nazari, Mohammad Hasan Shahriari, Hamed Tabesh. BigData Analysis in Healthcare: Apache Hadoop, Apache spark and Apache Flink. Frontiers in HealthInformatics, 2019.
16. Diego García-Gil, Sergio Ramírez-Gallego, Salvador García, Francisco Herrera. A comparison on scalability for batch big data processing on Apache Spark and Apache Flink. Big Data Analytics, 2017.
17. Paris Carbone, Asterios Katsifodimos, Stephan Ewen, V. Markl, Seif Haridi, K. Tzoumas. Apache Flink™: Stream and Batch Processing in a Single Engine. IEEE Data Eng. Bull, 2015.
18. Pritika Bahad, Preeti Saxena. Study of AdaBoost and Gradient Boosting Algorithms for Predictive Analytics. Algorithms for Intelligent Systems, 2019.
19. Candice Bentéjac, Anna Csörgő & Gonzalo Martínez-Muñoz. A comparative analysis of gradient boosting algorithms. Artificial Intelligence Review, 2021.
20. Yifei Chen, Zhenyu Jia, Dan Mercola, Xiaohui Xie. A Gradient Boosting Algorithm for Survival Analysis via Direct Optimization of Concordance Index. Hindawi, 2013.
21. D. Alghazzawi, O. Bamasag, H. Ullah, and M. Z. Asghar,"Efficient detection of DDoS attacks using a hybrid deep learning model with improved feature selection," Applied Sciences, vol. 11, no. 24, p. 11634, Dec. 2021.
22. Kerneler. ACK/PUSH-ACK DDoS Dataset. Available online: https://www.kaggle.com/datasets/yashwanthkumbam/apaddos-dataset (accessed on 7 August 2022).
23. Tilmann Rabl*, Jonas Traub, and Volker Markl, 'Apache Flink in Current Research Projects' 304536933 Researcher.net
24. N. Deshai, B.V.D.S. Sekhar, S. Venkataramana, 'Processing Big Data with Apache Flink',International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8 Issue-1S3, June 2019.
25. Dr. Yusuf Perwej , 'A Comprehend the Apache Flink in Big Data Environments', 2018, IOSR Journal of Computer Engineering (IOSR-JCE), e-ISSN: 2278-0661, P-ISSN: 2278-8727
26. Panpan Qi, Wei Wang, Lei Zhu, and See Kiong Ng. 2021. Unsupervised Domain Adaptation for Static Malware Detection based on Gradient Boosting Trees. In Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM '21), November 1–5, 2021, Virtual Event, QLD, Australia. ACM, New York, NY, USA, 10 pages. M. Shurman, R. Khrais, and A. Yateem, "DoS and DDoS attack detection using deep learning and IDS," The International Arab Journal of Information Technology, vol. 17, no. 4A, pp. 655–661, 2020.
27. Shurman, R. Khrais, and A. Yateem, "DoS and DDoS attack detection using deep learning and IDS," 0e International Arab Journal of Information Technology, vol. 17, no. 4A, pp. 655–661, 2020.
28. Nugraha, N. Kulkarni, and A. Gopikrishnan, "Detecting adversarial DDoS attacks in software- defined networking using deep learning techniques and adversarial training," 2021 IEEE International Conference on Cyber Security and Resilience (CSR), in Proceedings of the 2021 IEEE International Conference on Cyber Security and Resilience (CSR), pp. 448– 454, Rhodes, Greece, July. 2021.
29. Zhang, H.; Dai, S.; Li, Y.; Zhang, W. Real-time distributed-random-forest-based network intrusion detection system using Apache spark. In Proceedings of the 2018 IEEE 37th International Performance Computing and Communications Conference (IPCCC), Orlando, FL, USA, 17–19 November 2018; pp. 1–7.
30. Syed, N.F.; Baig, Z.; Ibrahim, A.; Valli, C. Denial of service attack detection through machine learning for the IoT. J. Inf. Telecommun. 2020, 4, 482–503.
31. Hu, T., Li, X., and Zhao, Y. (2007). "Gradient boosting learning of Hidden Markov models," in Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'06) (Toulouse). doi: 10.1109/ICASSP.2006.1660233