Efficient model for detecting application layer distributed denial of service attacks

Morenikeji Kabirat Kareem[1], Olaniyi Dada Aborisade[1], Saidat Adebukola Onashoga[1], Tole Sutikno[2], Olaniyi Mathew Olayiwola[3]

[1]Department of Computer Science, College of Physical Sciences, Federal University of Agriculture, Abeokuta, Nigeria
[2]Department of Electrical Engineering, Faculty of Industrial Technology, Universitas Ahmad Dahlan, Yogyakarta, Indonesia
[3]Department of Statistics, College of Physical Sciences, Federal University of Agriculture, Abeokuta, Nigeria

Article Info

Article history:

Received Mar 31, 2022
Revised Aug 18, 2022
Accepted Oct 10, 2022

Keywords:

Application layer k-nearest neighbor
Distributed denial of service attack
Principal component analysis
Support vector machine
Web-based service
XGBoost

ABSTRACT

The increasing advancement of technologies and communication infrastructures has been posing threats to the internet services. One of the most powerful attack weapons for disrupting web-based services is the distributed denial of service (DDoS) attack. The sophisticated nature of attack tools being created and used for launching attacks on target systems makes it difficult to distinguish between normal and attack traffic. Consequently, there is a need to detect application layer DDoS attacks from network traffic efficiently. This paper proposes a detection system coined eXtreme gradient boosting (XGB-DDoS) using a tree-based ensemble model known as XGBoost to detect application layer DDoS attacks. The Canadian institute for cybersecurity intrusion detection systems (CIC IDS) 2017 dataset consisting of both benign and malicious attacks was used in training and testing of the proposed model. The performance results of the proposed model indicate that the accuracy rate, recall, precision rate, and F1-score of XGB-DDoS are 0.999, 0.997, 0.995, and 0.996, respectively, as against those of k-nearest neighbor (KNN), support vector machine (SVM), principal component analysis (PCA) hybridized with XGBoost, and KNN with SVM. So, the XGB-DDoS detection model did better than the models that were chosen. This shows that it is good at finding application layer DDoS attacks.

Corresponding Author:

Olaniyi Dada Aborisade
Department of Computer Science, College of Physical Sciences, Federal University of Agriculture
Abeokuta, Nigeria
Email: aborisadeda@funaab.edu.ng

1. INTRODUCTION

It is no longer news that our daily activities are significantly dependent on the internet, backed by significant advancements in technological and communication infrastructures [1]. The general advancement in technologies and developments in network infrastructures in particular have attracted many users to these technologies, including malicious users who pose threats to computer networks and cyber systems, leading to an increased incidence of cyber-attacks [1]. Prominent among these cyber-attacks is the distributed denial of service (DDoS) attack, which has been observed to be a significant weapon of cyber-attacks in recent times. DDoS is essentially used to overload network resources such as memory and bandwidth, lowering computer network performance [2]. It is regarded as a hazardous attack on network security [2], [3]. A DDoS attack is a category of attack involving multiple or groups of devices attacking an underlying server to exhaust the network resources and legitimate users are denied access to network services. DDoS attacks are designed primarily to deprive legitimate users of resources and render network services unavailable to them [3], [4].

The major targets of attackers are the transport and network layers, but as technology advances, the application layer is now susceptible to DDoS attacks. DDoS attacks on application layer crash and exhaust the network server's resources, make the service unavailable to a legitimate host connected to the server, or infect the connected host through the exploitation of application layer protocols [5], [6]. Slowloris and hypertext transfer protocol (HTTP) flooding are examples of this application layer attack. An HTTP flood attacks web servers by utilizing HTTP GET or POST requests [6]. Slowloris attacks the web server by sending a partial HTTP request to the targeted server, causing the targeted server to open additional connections [6]. Figure 1 depicts the typical architecture of a DDoS attack where the attacker indirectly accesses the agent through handlers; the attacker could access many possible agents and handlers needed for launching a DDoS attack. The agent is responsible for sending many useless packets to the target victim simultaneously where the network resources are exhausted and the service availability is shut down [7]–[9].

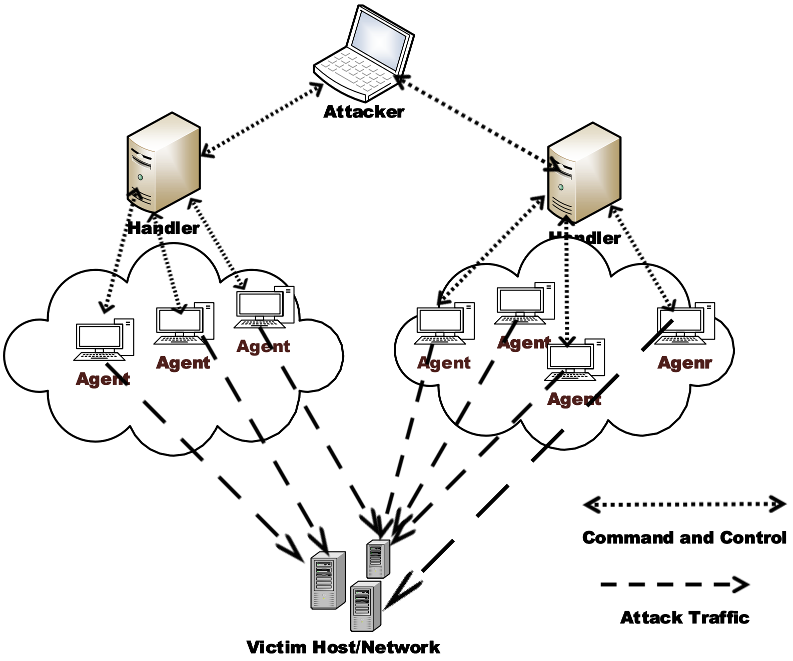

Figure 1. Typical DDoS architecture [7], [10]

Application DDoS attacks are web-based activities, and when they occur, all application services are put on hold [10], [11]. As these attacks increase drastically with sophisticated tools [12], [13] web application layer needs more security attention. Identifying an application layer DDoS attack is more difficult because it uses a fake address known as a spoofed IP to launch its attacks, thereby making a trace to the source of the attack uneasy [14]–[16]. Consequently, there is a need for efficient security of web services from DDoS attacks to accurately classify the network traffic as normal or attack traffic. Many approaches have been adopted for classifying DDoS attacks on network and web systems. Many of these approaches include statistical methods that are usually referred to as traditional methods. These traditional methods have been found insufficient for detecting DDoS attacks; hence, machine learning techniques are now being adopted. Consequently, machine learning-based DDoS detection has received increased attention in recent times. Some of the recent machine learning-based papers found in [17]–[21], were reviewed and described as follows. Alhayali *et al.* [17] a number of machine learning optimizing algorithms were combined for feature selection and weighting and feature subset selection (FSS). Furthermore, multi-objective optimization was adopted to choose the fewest characteristics without compromising FSS accuracy. The suggested Rao-algorithm-specific, support vector machine (SVM) parameter-less idea was also examined in the work while KDDCup 99 and Canadian institute for cybersecurity intrusion detection systems (CICIDS) 2017 datasets were employed. On the KDDCup 99 dataset, rao-SVM was more accurate than other methods by 100%, and on the CICIDS dataset, it was more accurate by 97.5%.

In another related work, a machine learning-dependent approach to detect DoS attacks was proposed in a client-server environment. A dataset of traffic data and DoS attacks were fed into their proposed algorithm for training. The trained algorithm was able to identify DoS attacks from other network traffic packets. Consequently, a very high percentage of right classifications was achieved by [18]. In order to address the problems of intrusion and cyber-attacks in network systems with single classifiers, [19] presented a majority voting-based ensemble model capable of being used in real-time to successfully examine network traffic and preemptively inform against possible attacks of some popular DoS assaults. The proposed model was found very effective for detecting intrusions on network systems [19]. Rajagopal *et al.* [20] used the stacking idea to provide an ensemble solution for network intrusion detection. The suggested method was implemented using Graphlab to present a powerful processing paradigm that handled large amounts of data. To show the reliability of predictions, two benchmark datasets-UNSW NB-15 and UGR '16-were used to evaluate and validate the prediction outputs. The experiment results showed that the performance of the proposed technique was excellent [20]. The need for real-time detection of aberrations of network packets through the adoption of machine learning classifiers to learn and classify compromised ones from uncompromised ones between their source and destination was tested [21]. The proposed detection model was evaluated and found applicable for real-time detection, especially when sensitive information is involved.

In the reviewed paper, intrusion detection using either optimized or basic binary classification machine learning was mostly adopted. In these, intrusion detection was either applied to a general network system or a DoS, but this paper focuses on detecting DDoS attacks on the application layer. This paper is intended to develop a model for detecting application layer DDoS attacks using a tree-based ensemble model known as eXtreme gradient boosting (XGBoost) and XGBoost combined with principal component analysis (PCA); and compare their performances with selected machine learning models [22]. The XGBoost algorithm has been regarded as an outstanding performer for solving classification problems due to its scalability in all scenarios, speed, and accuracy [23]. This paper adopts XGBoost with and without the PCA for detecting application layer DDoS attacks [24].

2. METHOD

This section discusses the proposed model, coined XGB-DDoS, and the architecture for detecting application DDoS attacks. A tree-based model called XGBoost is proposed to classify malicious and non-malicious traffic efficiently. Figure 2 depicts the proposed detection architecture where legitimate and illegitimate user requests are sent to the web server through the internet. The XGB-DDoS inspects requests to verify if they are malicious or not. If the traffic is malicious, the XGB-DDoS detects the attack before it enters the server.

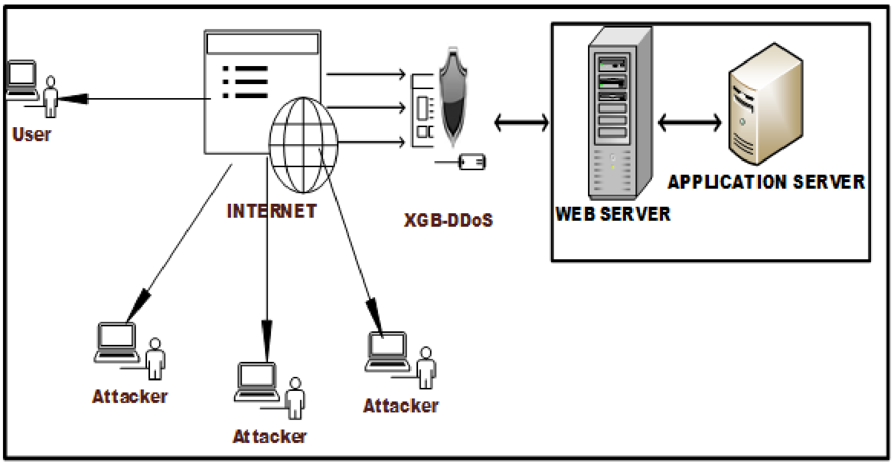

Figure 2. XGB-DDoS architecture

2.1. eXtreme gradient boosting

XGBoost is a widely used supervised machine learning algorithm for classification and regression. Because of its scalability, data scientists widely use it to solve many machine learning challenges and real-world scale problems. It is a new version of the gradient boosting decision tree (GBDT) that supports distributed and parallel computation, making the model's training faster and having higher execution time and cache optimization. The main advantages of the XGBoost algorithm are speed, scalability, and high-performance using gradient-boosted decision trees [25].

It is approximately ten times faster than the previous methods on a single platform, eradicating time consumption, particularly during preprocessing. It outshines other libraries by allowing tuning of regularization parameters, and it was recorded as a winning model published on Kaggle's blog in 2015 with 17 solutions [26]. The XGBoost addresses overfitting as well as other classification-related issues. It predicts by combining weak base learners to form stronger learners, as depicted by (1). The XGBoost model is trained in a way that adds up [26], so that (1) gives the prediction.

$$\hat{\mathrm{z}}_i = f(\mathrm{y}_i) = \sum_{j=1}^{M} f_j(y_i) \tag{1}$$

Let $f_j(y_i)$ denote a base learner function, M the number of base learners, and $\hat{\mathrm{z}}_i$ represents prediction at the i-th sample. XGBoost emerged to solve the problem of overfitting and some other classification task-related problems [26]. As shown in (2), the regularized objective function must be minimized to learn the set of functions used in the model. The objective function is to monitor the performance of the model during training.

$$obj(\emptyset) = [TL(\emptyset) + \Omega(\emptyset)\ ] \tag{2}$$

$\mathrm{T}L(\emptyset) = l(\mathrm{z}_i, \hat{\mathrm{z}}_i)$, this is the training loss, while $\Omega(\emptyset)$ is the regularization for penalizing the complexity of the model to avoid overfitting (3).

$$L(\emptyset) = \sum_{i=1}^{M} l(z_i, \hat{\mathrm{z}}_i + \varphi_t(x)) + \ \Omega(\emptyset) \tag{3}$$

XGBoost uses a Newton boosting method [26], requiring the loss function to be twice differentiable to optimize the objective function quickly. In training the model, the optimal weight of the leaf is calculated for a fixed structure of the tree and enumerated for all possible trees. Then, pick the best one that optimizes the objective function, i.e., splits the leaf into two leaves on the new left and right leaf, and see the score it gains. Predictors are made dependently and sequentially, and the subsequent predictors learn from the mistakes of the previous predictor.

2.2. Principal component analysis

PCA is known as an unsupervised learning and dimension-reduction technique. When there is a large dataset, it is usually difficult to interpret. PCA is used to reduce the dimension of the dataset with a high number of features while preserving as much relevant information as possible and minimizing information loss [27]. PCA is a data exploratory analysis method that reduces data dimensionality. It enables less data storage space, noise reduction, and collinearity removal. The original data of possibly correlated variables is transformed into linear uncorrelated (PCA) data set values [27].

2.3. Dataset

In this work, a well-labeled Canadian dataset, the CIC IDS 2017 dataset [28]. It includes both benign and malicious attacks used in training and testing the proposed system. The data was captured and recorded on Wednesday, July 5th, 2017. The dataset consists of the following advanced types of attacks (DOS Slowloris, DOS Hulk, DOS Slowhttptest, and DOS GoldenEye): some of the primary application-layer DDoS attacks. These datasets contain 80 extracted features with a total number of 692,703 flows. Table 1 shows the types of attacks present in the dataset used in this work. Table 2 depicts the total number of rows in the dataset.

Table 1. CIC IDS dataset analysis

S/n	Traffic type	Number
1	Benign	440,031
2	DoS Slowloris	5,796
3	DoS Hulk	231,073
4	DoS Slowhttptest	5,499
5	DoS Goldeneye	10,293
6	Heartbleed	11
	Total	692,703

Table 2. Data set used for the experiment

Category	Training data set	Testing dataset
Benign	426,822	13,209
Dos attacks	176,870	75,802

2.4. Dataset pre-processing

As depicted in Figures 3 and 4, the dataset was first preprocessed to fit in for training the model. In the dataset, all the rows with not a number (NAN) were dropped. The data set has a multi-class label, and it is essential to transform it into a numerical value that the machine understands. A label binarizer was adopted for data transformation. A label a binarizer is a technique in machine learning for converting multi-class to binary labels. XGBoost can be trained without standardizing the dataset since it is a tree-based model, but data standardization is essential when combined with PCA. Data standardization is a method of rescaling features to a mean value of 0 and a standard deviation of 1. In PCA, features with high variance usually have priority over features with low variance. A Standard Scalar technique was used on the data set for rescaling to prevent inadequacy. The rescaled data is fed into the PCA algorithm for dimensionality reduction, which helps to abolish redundant features from the data set. In training and testing the model, the dimensionally reduced data was fed into the XGB classifier. Figure 3 depicts the steps involved in the XGB-DDoS system; Figure 4 depicts the hybridized system using the XGBoost and PCA. Algorithm 1 explains how the XGB-DDoS model detects DDoS attacks and non-attacks.

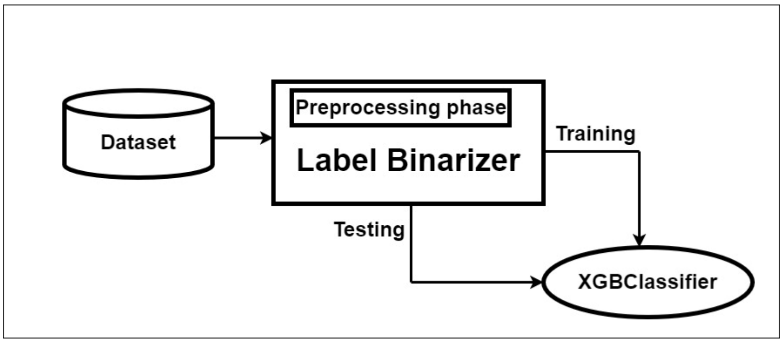

Figure 3. XGB-DDoS process

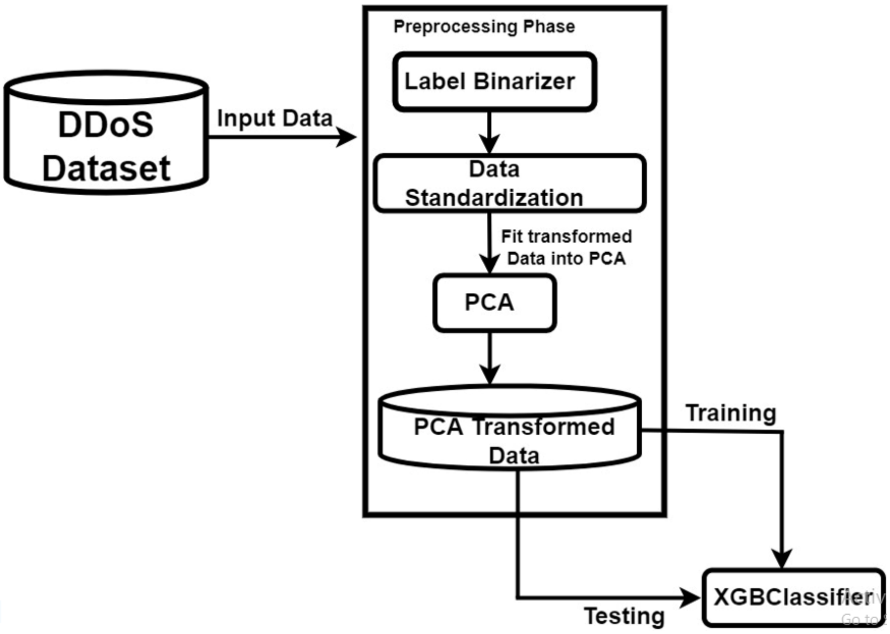

Figure 4. XGB-PCA process

Algorithm 1. XGB-DDoS model

```
INPUT: XGB-PCA, Data set sample, $D_{sample}$
OUTPUT: DDoS attack, non-attack
PROCESS:
$flag \leftarrow$ XGB − DDoS ($D_{sample}$)
if flag == non_DDos_attack then
    forward request
else if flag == DDoS attack then
    Request Access denied
end if
end
```

3. RESULTS AND DISCUSSION

The experiment was carried out on an Intel Core i5 computer with 8G RAM and a terabyte disk. The model was developed using the Python sci-kit-learn library with jupyterlab. The CIC IDS dataset is well-labeled, where 70% of the dataset was adopted for training and 30% for testing the model. The experimental information provided by the confusion matrix as represented in Table 3 was used to calculate the accuracy, precision, recall, and F1 scores, used to evaluate the models.

- Accuracy: this is referred to as the rate at which the system correctly distinguish between the DDoS attack and non- attack that is percentage correctly classified, it is estimated with (4).

$$Accuracy = (TP + TN) / (TP + TN + FP + FN) \tag{4}$$

- Precision: this is the correct rate of DDoS attack detected by the system; a better system should possess a higher precision rate and is estimated with (5).

$$Precision = \frac{TP}{TP+FP} \tag{5}$$

- Recall or sensitivity: this rate that shows the proportion of True Positive rate was classified by the detection model as DDoS attack, is estimated with (6).

$$\text{Recall or Sensitivity} = \text{TP}/ (\text{TP} + \text{FN}) \tag{6}$$

- F1 score: this measure of the detection model accuracy it is the harmonic mean of precision and recall, is estimated with (7).

$$F1\ score = 2x\ (Precision\ x\ Recall)/\ (Precision\ Recall) \tag{7}$$

Table 3. Confusion matrix for the XGB-DDoS

Predicted label	Confusion matrix					
Benign	32,004	15	14	11	14	0
Goldeneye	16	3,061	0	1	1	0
Hulk	122	19	69,155	0	0	0
Slowhttptest	9	1	0	1,619	0	0
Slowloris	0	0	0	2	1,746	0
Heartbleed	0	0	0	0	0	1

The result of the proposed model was compared with SVM and k-nearest neighbour (KNN) in terms of accuracy, recall, F1-Score and precision. The performance results of each algorithm with PCA were recorded. XGBoost without the PCA is coined XGB-DDoS, while XGBoost with the PCA is XGB-PCA. From the experimental results, XGB-PCA and XGB-DDoS have an accuracy rate of 0.997 and 0.999, respectively. Table 4 shows the overall accuracy performance of the selected algorithm. KNN, SVM, KNN-PCA, and SVM-PCA have the following accuracy rates: 0.990, 0.994, 0.999, and 0.995, respectively.

This result shows that the XGBoost algorithm alone is sufficient and has a high-performance rate; without the inclusion of PCA, it can handle missing data and multi-class labels. Table 4 shows the overall accuracy performance of the selected algorithms. KNN, SVM, KNN-PCA, and SVM-PCA have the following accuracy rates: 0.990, 0.992, 0.994, and 0.994, respectively. The accuracy performance increased with the addition of PCA, while there was no improvement in SVM. Nonetheless, the accuracy metric is not enough to measure the machine learning model's performance, and other performance metrics are displayed with the bar charts. Figure 5 shows the performance rate of XGB-PCA and XGB-DDoS in terms of recall, precision, and f1-score. From the experimental results, PCA has no significant contribution to the XGBoost algorithm. Without PCA, XGBoost outperformed XGB-PCA; as a result, XGBoost is an excellent classifier.

Table 4. Accuracy performance between algorithms

Algorithms	Accuracy rate
XGB-DDoS	0.999
XGB-PCA	0.997
KNN	0.990
KNN-PCA	0.999
SVM	0.994
SVM-PCA	0.995

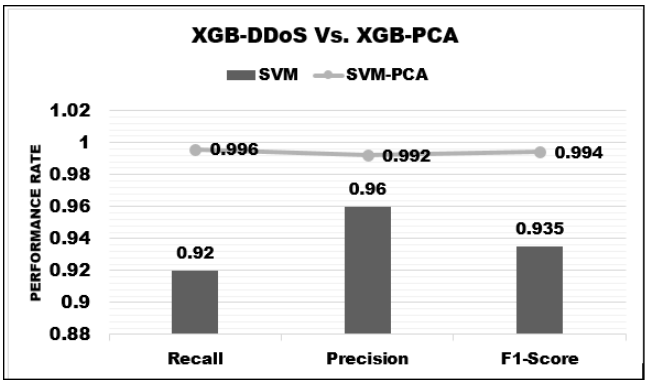

Figure 5. Performance comparison between XGB-DDoS and XGB-PCA

The performance of XGB-PCA and XGB-DDoS in terms of recall, precision, and F1-score rate is: 0.996, 0.992, 0.994, and 0.997, 0.995, 0.996, respectively. Figure 6 shows the performance of KNN and KNN with PCA. The recall, precision, and F1-score rate of KNN and KNN with PCA (KNN-PCA) are: 0.984, 0.916, 0.944, and 0.993, 0.995, 0.979, respectively. From the result, the impact of PCA is obvious, and it improved the performance of KNN; hence, KNN-PCA outperformed ordinary KNN. Figure 7 is the bar chart showing the performance results of SVM and SVM with PCA (SVM-PCA). There is a conspicuous performance difference between SVM and SVM-PCA in terms of recall and F1-score, but a slight difference of 0.001 for precision. In the chart, SVM is shown to have a rate of 0.920, 0.960, and 0.935 for recall, precision, and f1-score, respectively. While the rate of recall, precision and recall for SVM with PCA are 0.951, 0.961, and 0.979 respectively. From the analysis of the experiment, it can be seen that PCA has a big effect and greatly improves the performance of traditional machine learning algorithms. However, the performance of the XGB-DDoS does not improve in a noticeable way.

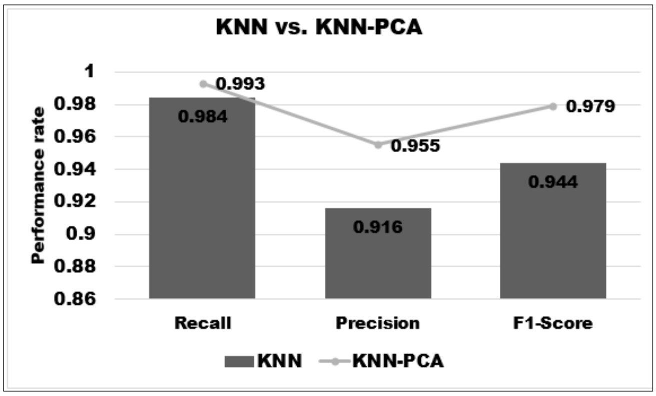

Figure 6. Performance comparison between KNN and KNN-PCA

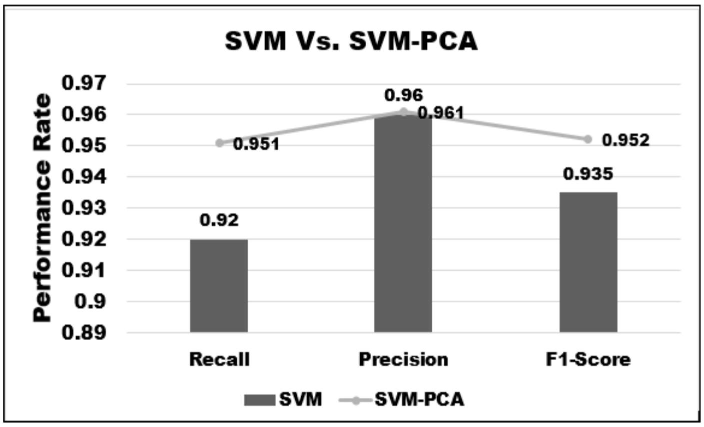

Figure 7. Performance comparison between SVM and SVM-PCA

4. CONCLUSION

The DDoS attack, a variant of the DoS attack, is no doubt one of the commonest and biggest security challenges to networks and computing systems in recent times. Although a number of recent research efforts have been geared towards addressing the problem of DoS attacks with simple machine learning techniques, intrusion detection using ensemble and majority voting approaches, and intrusion predictive models on general network and web service systems, none has presented a model for detecting a DDoS attack at the application layer. This paper developed efficient detection models for detecting application layer DDoS attacks. A standard Canadian dataset was employed to train and test the developed model, called XGB-DdoS. The experimental results show that the XGB-DDoS detection model outperformed other selected traditional machine learning models used in some recent DDoS detection research efforts. Hence, XGB-DDoS is efficient for detecting application DDoS attacks. Future directions of this work would be to incorporate a mechanism capable of blocking the attack after it has been detected.

REFERENCES

[1] G. S. Kushwah and V. Ranga, "Detecting DDoS attacks in cloud computing using extreme learning machine and adaptive differential evolution," *Wireless Personal Communications*, vol. 124, no. 3, pp. 2613–2636, Jun. 2022, doi: 10.1007/s11277-022-09481-9.

[2] A. Irum, M. A. Khan, A. Noor, and B. Shabir, "DDoS detection and prevention in internet of things," *EasyChair*, pp. 1–7, 2020.

[3] P. S. Saini, S. Behal, and S. Bhatia, "Detection of DDoS attacks using machine learning algorithms," in *2020 7th International Conference on Computing for Sustainable Global Development (INDIACom)*, Mar. 2020, pp. 16–21, doi: 10.23919/INDIACom49435.2020.9083716.

[4] S. Smadi, M. Alauthman, O. Almomani, and A. Saaidah, "Application layer denial of services attack detection based on stacknet," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, no. 3, pp. 3929–3936, Jun. 2020, doi: 10.30534/ijatcse/2020/215932020.

[5] I. Sreeram and V. P. K. Vuppala, "HTTP flood attack detection in application layer using machine learning metrics and bio inspired bat algorithm," *Applied Computing and Informatics*, vol. 15, no. 1, pp. 59–66, Jan. 2019, doi: 10.1016/j.aci.2017.10.003.

[6] G. A. Jaafar, S. M. Abdullah, and S. Ismail, "Review of recent detection methods for HTTP DDoS attack," *Journal of Computer Networks and Communications*, pp. 1–10, Jan. 2019, doi: 10.1155/2019/1283472.

[7] A. H. B. Alghuraibawi, R. Abdullah, S. Manickam, and Z. A. A. Alyasseri, "Detection of ICMPv6-based DDoS attacks using anomaly based intrusion detection system: A comprehensive review," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 6, pp. 5216–5228, Dec. 2021, doi: 10.11591/ijece.v11i6.pp5216-5228.

[8] M. S. Todorova and S. T. Todorova, "DDoS attack detection in SDN-based VANET architectures," *Master Appl Sci*. pp. 1–175, 2016.

[9] F. S. D. L. Filho, F. A. F. Silveira, A. D. M. B. Junior, G. Vargas-Solar, and L. F. Silveira, "Smart detection: An online approach for DoS/DDoS attack detection using machine learning," *Security and Communication Networks*, pp. 1–15, Oct. 2019, doi: 10.1155/2019/1574749.

[10] N. S. Vishnu, R. S. Batth, and G. Singh, "Denial of service: Types, techniques, defence mechanisms and safe guards," in *Proceedings of 2019 International Conference on Computational Intelligence and Knowledge Economy, ICCIKE 2019*, Dec. 2019, pp. 695–700, doi: 10.1109/ICCIKE47802.2019.9004388.

[11] M. M. Oo, S. Kamolphiwong, and T. Kamolphiwong, "The design of SDN based detection for distributed denial of service (DDoS) attack," in *2017 21st International Computer Science and Engineering Conference (ICSEC)*, Nov. 2017, pp. 258–263, doi: 10.1109/ICSEC.2017.8443939.

[12] S. Haider *et al.*, "A deep CNN ensemble framework for efficient DDoS attack detection in software defined networks," *IEEE Access*, vol. 8, pp. 53972–53983, 2020, doi: 10.1109/ACCESS.2020.2976908.

[13] N. Ahuja, G. Singal, and D. Mukhopadhyay, "DLSDN: Deep learning for DDOS attack detection in software defined networking," in *2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, Jan. 2021, pp. 683–688, doi: 10.1109/Confluence51648.2021.9376879.

[14] N. Hoque, H. Kashyap, and D. K. Bhattacharyya, “Real-time DDoS attack detection using FPGA,” *Computer Communications*, vol. 110, pp. 48–58, Sep. 2017, doi: 10.1016/j.comcom.2017.05.015.

[15] A. B. Dehkordi, M. Soltanaghaei, and F. Z. Boroujeni, “The DDoS attacks detection through machine learning and statistical methods in SDN,” *The Journal of Supercomputing*, vol. 77, no. 3, pp. 2383–2415, Mar. 2021, doi: 10.1007/s11227-020-03323-w.

[16] M. Asad, M. Asim, T. Javed, M. O. Beg, H. Mujtaba, and S. Abbas, “Deepdetect: Detection of distributed denial of service attacks using deep learning,” *The Computer Journal*, vol. 63, no. 7, pp. 983–994, Jul. 2020, doi: 10.1093/comjnl/bxz064.

[17] R. A. I. Alhayali, M. Aljanabi, A. H. Ali, M. A. Mohammed, and T. Sutikno, “Optimized machine learning algorithm for intrusion detection,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 24, no. 1, pp. 590–599, Oct. 2021, doi: 10.11591/ijeecs.v24.i1.pp590-599.

[18] M. M. Rasheed, A. K. Faieq, and A. A. Hashim, “Development of a new system to detect denial of service attack using machine learning classification,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 23, no. 2, pp. 1068–1072, Aug. 2021, doi: 10.11591/ijeecs.v23.i2.pp1068-1072.

[19] A. M. Bamhdi, I. Abrar, and F. Masoodi, “An ensemble based approach for effective intrusion detection using majority voting,” *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 19, no. 2, pp. 664–671, Apr. 2021, doi: 10.12928/telkomnika.v19i2.18325.

[20] S. Rajagopal, P. P. Kundapur, and H. K. Siddaramappa, “A predictive model for network intrusion detection using stacking approach,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 3, pp. 2734–2741, Jun. 2020, doi: 10.11591/ijece.v10i3.pp2734-2741.

[21] N. P. Shetty, J. Shetty, R. Narula, and K. Tandona, “Comparison study of machine learning classifiers to detect anomalies,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 5, pp. 5445–5452, Oct. 2020, doi: 10.11591/ijece.v10i5.pp5445-5452.

[22] S. Bhattacharya *et al.*, “A novel PCA-firefly based XGBoost classification model for intrusion detection in networks using GPU,” *Electronics*, vol. 9, no. 2, p. 219, Jan. 2020, doi: 10.3390/electronics9020219.

[23] S. Dhaliwal, A.-A. Nahid, and R. Abbas, “Effective intrusion detection system using XGBoost,” *Information*, vol. 9, no. 7, p. 149, Jun. 2018, doi: 10.3390/info9070149.

[24] F. Kherif and A. Latypova, “Principal component analysis,” in *Machine Learning*, Amsterdam: Elsevier, 2020, pp. 209–225, doi: 10.1016/B978-0-12-815739-8.00012-2.

[25] Z. Wen, B. He, R. Kotagiri, S. Lu, and J. Shi, “Efficient gradient boosted decision tree training on GPUs,” in *2018 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, May 2018, pp. 234–243, doi: 10.1109/IPDPS.2018.00033.

[26] D.-K. Choi, “Data-driven materials modeling with XGBoost algorithm and statistical inference analysis for prediction of fatigue strength of steels,” *International Journal of Precision Engineering and Manufacturing*, vol. 20, no. 1, pp. 129–138, Jan. 2019, doi: 10.1007/s12541-019-00048-6.

[27] B. M. S. Hasan and A. M. Abdulazeez, “A review of principal component analysis algorithm for dimensionality reduction,” *Journal of Soft Computing and Data Mining*, vol. 02, no. 01, pp. 20–30, Apr. 2021, doi: 10.30880/jscdm.2021.02.01.003.

[28] Canadian Institute for Cybersecurity, “Datasets,” *University of New Brunswick (UNB)*, 2022. https://www.unb.ca/cic/datasets/ (accessed Nov. 07, 2022).

BIOGRAPHIES OF AUTHORS

Morenike Kabirat Kareem received her Bsc. and Msc. degree in Computer Science from the Federal University of Agriculture, Abeokuta (FUNAAB). She is currently undergoing her PhD programme in Computer Science. Her research interest includes artificial intelligence and cybersecurity. She can be contacted at email: kareemmk@funaab.edu.ng.

Olaniyi Dada Aborisade received Bachelor of Science (Bsc.) Degree from University of Agriculture, Abeokuta, Nigeria in Mathematical Sciences (Computer Science option) in year 2000, Masters (MSc) in Degree of the University of Ibadan, Nigeria in 2007. He undertook part of his Ph.D research benchwork at Hochschule Furtwangen University, (HFU) Germany between 2015 and 2016, and completed Ph.D in Computer Science in 2017 at the Federal University of Agriculture, Abeokuta, Nigeria. He is a Lecturer in the Department of Computer Science, College of Physical Sciences, Federal University of Agriculture Abeokuta (FUNAAB), Nigeria. He is a member of both local and international professional bodies. His research interests include cloud security, digital forensics, and using artificial intelligence. He can be contacted at email: aborisadeda@funaab.edu.ng.

Saidat Adebukola Onashoga received Bachelor of Science (Bsc.) degree in Mathematical Sciences (Computer Science option), Masters (MSc) in degree, from University of Agriculture, Abeokuta and PhD degree of the Federal University of Agriculture, Abeokuta, Nigeria in Computer Science 2000, 2003, and 2010 respectively**.** She is a Professor in the Department of Computer Science, College of Physical Sciences, Federal University of Agriculture, Abeokuta, Ogun State, Nigeria. She has publications in both local and international journals. She can be contacted at email: onashogasa@funaab.edu.ng.

Tole Sutikno is a lecturer in the Electrical Engineering Department at the Universitas Ahmad Dahlan (UAD), Yogyakarta, Indonesia. He received his B.Eng., M.Eng., and Ph.D. degrees in Electrical Engineering from Universitas Diponegoro, Universitas Gadjah Mada, and Universiti Teknologi Malaysia, in 1999, 2004, and 2016, respectively. He has been an Associate Professor at UAD, Yogyakarta, Indonesia since 2008. He is currently the *Editor-in-Chief* of the Bulletin of Electrical Engineering and Informatics and the Head of the Embedded Systems and Power Electronics Research Group. His research interests include the fields of digital design, industrial applications, industrial electronics, industrial informatics, power electronics, motor drives, renewable energy, FPGA applications, embedded systems, artificial intelligence, intelligent systems, information systems, and digital libraries. He can be contacted at email: tole@ee.uad.ac.id.

Olaniyi Mathew Olayiwola is a Reader in the Department of Statistics, College of Physical Sciences, Federal University of Agriculture, Abeokuta, Ogun State, Nigeria. He obtained National Diploma (ND) in Statistics in 1995, Higher National Diploma (HND) in Statistics in 1998, Postgraduate Diploma in Statistics in 2002, Postgraduate Diploma in Education in 2006, Bachelor of Science (B.Sc.) degree in Statistics in 2010 from University of Ilorin and Master of Science (M.Sc.) in Statistics from the University of Ibadan in 2005. He obtained Doctor of Philosophy (Ph.D) degree in Statistics at the University of Ibadan Nigeria in 2011. He can be contacted at email: olayiwolaom@funaab.edu.ng.