International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(9s) 2026 pp. 830-840
https://doi.org/10.70917/ijcisim-2026-3505
Article

Assessment of Ensemble Bagging for Robust and Adaptive Identification of DDoS Attacks in Software-Defined Networks

Sudhir Bhagat[1*], Himanshu Gupta[2], Ashish Seth[3]

[1]Amity Institute of Information Technology (AIIT), Amity University, Noida (U.P) - 201313, India
[2]Amity Institute of Information Technology (AIIT), Amity University, Noida (U.P) - 201313, India
[3]Centre for Research & Development, Sri Eshwar College of Engineering, Coimbatore, Tamil Nadu - 641202, India
Email: [1*]research.sb20@gmail.com

Abstract: The increasing frequency and severity of Distributed Denial-of-Service (DDoS) attacks, which overload resources and block authorized access, pose a serious threat to the stability of modern networks. Robust and flexible security measures are needed to counter these threats. The incapacity of conventional protocols to adjust to changing threats is a problem that machine learning (ML) offers a workable solution to. Because ML algorithms can automatically detect fraudulent activity and improve accuracy, they are especially helpful in software-defined networks (SDN), where centralized controllers are very sensitive. This work explores machine learning-based methods for detecting DDoS attacks in SDN, with a focus on bagging decision tree algorithms and random forest ensemble technology. Performance was measured using accuracy, relative absolute error (RAE), and relative squared error (RRSE). The results show that bagging consistently outperforms random forest and yields higher detection efficiency. The proposed model unexpectedly outperforms existing benchmarks with an accuracy rate of 99.94%, suggesting that it could enhance SDN security.

Keywords: Bagging Algorithm, Distributed Denial Of Service Attacks (DDOS), Ensemble Algorithm, Random Forest Algorithm, Software Defined Networking.

1. INTRODUCTION

Numerous innovative and inventive things have been produced by modern technology. One such trend is Software-Defined Networking (SDN). Because of its intricate network architecture and automated procedures it enhances the network system. A computer program called a network controller (SDN controller) manages a network. Both manual and automated methods can be used for this. Separating control and command execution, 02 crucial components of the network is its central concept [1].

Network architecture hasn't changed much over the last few decades which has made it more complicated. Large volumes of data must be sent by modern programs which are easily replaceable. For this reason the sophisticated networking technique known as Software-Defined Networking (SDN) is a good option. By separating the systems data and control components this approach facilitates system modification. Separate the components that manage data-transmitting devices such as switches, routers and modems to make things simpler. The network layer devices become straightforward forwarding devices when the data and control planes are divided. Every packet destination is decided by the controller located in the middle of the network. The networks rules and structure are drastically altered by these modifications. The heart and brain of SDN are comparable to its controller.

Unfortunately, this technology is now a prime target for malicious actors. DDoS attacks are the most dangerous because they have the ability to seriously disrupt systems [2]. Early detection and prevention of these attacks is critical to safeguard systems from harm. A software-defined networks ability to recognize and stop DDoS attacks is crucial in the current security environment. One common kind of network attack is DDoS attacks. It functions to oversee several

robot systems that transmit various network design-related issues to a single target [3]. The DDoS attack destroys the server's functionality making it impossible for regular users to utilize the target hosts services [4]. Currently detecting attacks and responding to attacks are the two primary methods of preventing DDoS attacks on a typical network [5].

Attack detection models have been developed extensively using machine learning (ML) [6]. In a software-defined network this can assist in detecting DDoS attacks. Digital age security issues can be resolved by machine learning algorithms [7]. By providing context-based features machine learning techniques assist systems in identifying patterns in large amounts of data and improving their current configuration. The system's ability to detect malicious activity can be improved by using machine learning algorithms to diagnose DDoS attacks.

Additionally machine learning models can offer defenses to lessen the impact of attacks. One can accomplish this by looking for trends and learning from the attacks prior data [8]. New attack techniques can be found and stopped by using machine learning algorithms.

DDoS attacks at the network layer which made up 71% (i.e. 59 lakh DDoS attacks) of all DDoS attacks that took place in the 3rd quarter of 2025 a rise of 95% year-over-year and 87% quarter-over-quarter. Nevertheless, only 29% of attacks were HTTP DDoS (i.e. 2.4 million DDoS attacks) of all DDoS attacks that took place in the 3rd quarter of 2025 dropped by 17% annually and 41% quarter-over-quarter[9]. The volume and intensity of distributed denial-of-service attacks reached previously unheard-of levels in year 2025 and the overall number of DDoS attacks observed more than doubled to 47.1 million according to a global network and cloud infrastructure platform, recently released threat first report of year 2026 [10].

We require robust security measures to stop these attacks because they are so potent. Attackers can be stopped and harmful activities like DDoS attacks can be detected by combining machine learning with more traditional security techniques. Software-defined networks are the most suitable settings for applying these methods to detect malicious activity on a network. This research aims to detect DDoS attacks in software-defined networks. The objective is to create a machine learning model that can recognize dangerous behaviors and determine the best course of action to avoid them.

2. IMPORTANCE OF THE RESEARCH

Distributed denial-of-service (DDoS) attacks are one of the most common and dangerous types of cyber threats. These attacks may make it difficult for legitimate users to access the service because they can overload system resources. In the context of software-defined networking (SDN), this issue is made even more critical by the centralized design of the SDN controller. The controller uses the OpenFlow protocol to manage the control plane while monitoring and directing network traffic [11]. It is the main target of attackers due to its central location, and once it is compromised, the entire network infrastructure may become unstable. DDoS attacks typically involve a master and several bots that inadvertently send millions of malicious requests to the target. This can cause major service disruptions and be abused for illegal or immoral purposes.

It is still vital to conduct research on DDoS attack detection in SDN. Many detection techniques have been documented, but in dynamic network environments, many of them lack accuracy, scalability, and adaptability. Ensemble learning techniques, especially bagging, offer a promising solution by reducing the variance of base classifiers and increasing their robustness. However, because bagging has not been extensively used in SDN-based DDoS detection, there is a sizable research gap. This study attempts to overcome that constraint by evaluating the performance of hybrid bagging algorithms using REPTree (Reduced-Error Pruning Tree) classifiers. Following the creation of multiple decision trees, REPTree selects the best tree. When used with bagging, ensemble variety can improve detection performance.

The purpose of this study is to determine whether ensemble bagging can provide a reliable, scalable, and accurate technique for identifying DDoS attacks in the SDN. The goal of this project is to increase SDN security through the use of REPTree classifiers in a hybrid approach. It offers a cutting-edge strategy based on machine learning to mitigate one of the main threats to modern network infrastructure.

3. REVIEW OF LITERATURE

SDN has completely changed how networks are run and managed. SDN is more scalable and adaptable than conventional networks. DDoS attacks on the other hand pose a security risk to the network that SDNs add. DDoS attacks overload the network by flooding it with malicious traffic. These assaults can be recognized and stopped by machine learning.

Fatmah Alanazi et al. (2021) [12] Authors suggests an ensemble detection framework based on deep learning (DL). By combining CNN LSTM and GRU architectures using three ensemble techniques four hybrid models were created. Using a limited set of flow-based features testing on the CICIDS2017 dataset revealed that the ensemble approach achieved 99.77% detection accuracy. This outcome shows that the suggested solution not only performs better than conventional techniques but also improves the SDNs availability security and flexibility making it a potent defense against DDoS attacks.

A machine learning method for identifying DDoS attacks in SDN was tested by Ahuja et al. (2021) [13]. To improve detection efficiency the researchers employed a hybrid model that combined features of support vectors and random forests. As a result there were no false alarms and the tests accuracy was 98.8%.

Tonkal et al. (2021) [14] concentrated on the detection of DDoS attacks in SDN using machine learning techniques in conjunction with Neighborhood Component Analysis (NCA), Artificial neural networks (ANN), support vector machines (SVM) and K-nearest neighbor (KNN) decision trees all had a 100% accuracy rate.

C. Srinivas et al. (2023) [15] Authors emphasizes how the centralized controller makes SDNs susceptible to DDoS attacks. A dataset of over 100000 flow records with 23 statistical features was used to test a number of machine learning algorithms for traffic classification. The Ensemble Random Forest algorithm outperformed KNN Decision Tree SVM and Random Forest with an accuracy of 99.99%. This outcome shows that Ensemble Learning offers a very accurate scalable and dependable detection mechanism guaranteeing enhanced security availability and flexibility in next-generation SDN environments.

In order to automatically identify DoS and DDoS attacks in the SDN, Elubeyd & Yiltas-Kaplan et al. (2023) [16] employed a deep learning model. Using accuracy rates of 99.81% and 99.88 % for two distinct datasets the authors examined six alternative models.

T. E. Ali et al. (2025) [17] Using genetic algorithm wrapper feature selection (GAWFS) which minimizes redundancy and chooses correlated features this study tackles the difficulties of identifying DDoS attacks. When used in conjunction with the stacking ensemble model (MLP SVM RF) it improved detectability by 40% and attained 99.86% training accuracy and 98.89% test accuracy. Its effectiveness and resilience on the benchmark dataset are demonstrated by the roughly 30% reduction in training time.

Makuvaza et al. (2021) [18] introduced a hybrid deep learning technique to identify DDoS attacks in FON (function-oriented networking). After comparing and analyzing two distinct approaches using three-character SVM and five-character random forest the authors discovered that the accuracy was 100%.

Alwabisi et al. (2022) [19] proposed an SDN-based and machine learning approach to stop DDoS attacks in fiber-optic networks. The Support Vector Machine (SVM) had three features making it 100% accurate according to the author's analysis of six machine learning models. The model took just 14.3 seconds to train.

DDoS attacks can be identified by software-defined networks using machine learning models and the reliability of algorithms is demonstrated through testing and comparison. Machine learning models lessen processing load and latency while increasing detection accuracy. This makes SDN a more attractive option for DDoS attack detection.

The aforementioned finding indicates that machine learning models are capable of identifying up to 98.8% of DDoS attacks. By removing even the smallest chance of misidentification the current study aims to increase DDoS attack accuracy. According to a review of the literature one model has been used extensively to identify DDoS attacks in SDN. We will employ a group approach based on machine learning to solve this issue.

4. RESEARCH METHODOLOGY

This study made use of the 23 rows and 104,345 columns of the DDoS SDN secondary dataset. It has 20 numerical attributes and is divided into 03 groups. The dataset contains a labeled target variable that is the basis for the classification. This numerical variable has a value between 0 and 1, where 1 represents benign traffic and 0 represents malicious traffic.

Its main objective is to accurately classify network traffic as malicious or legitimate using machine learning-based ensemble techniques. To do this, the dataset was examined using the WEKA software environment, which was chosen because of its many machine learning options and experimentation flexibility. Within this framework, two ensemble methods: Bagging and Random Forest were applied and compared.

The effectiveness of Bagging, which reduces variation by combining multiple classifiers, and Random Forest, which creates decision trees using randomized feature selection, in detecting DDoS traffic patterns was evaluated. The study compares these methods in order to determine which ensemble techniques produce more reliable and accurate results when identifying detrimental behavior in the SDN. This comparison shows how security against one of the most persistent cyber threats in modern networking can be enhanced by ensemble learning.

Each step of the bagging ensemble approach process data sampling, model training, aggregation, and final prediction output is clearly visible in Figure 1.

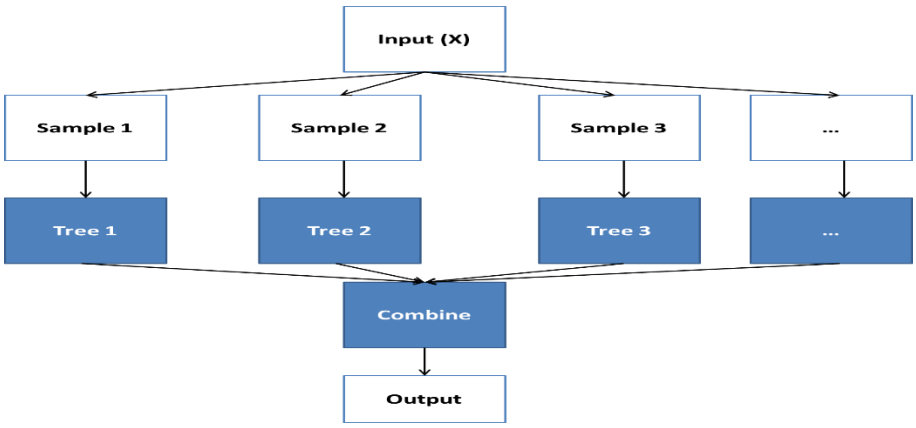


Figure 1: Bagging Ensemble

In an increasingly common machine learning technique known as ensemble learning, a number of separate models referred to as base models are combined to create an efficient optimal forecasting model. One instance of ensemble learning is the random forest method.

One method of organizing items for classification is bagging also known as bootstrap aggregation. Bagging algorithms are used in regression and classification models, particularly decision tree approaches, to avoid data overfitting. The bagging algorithm reduces variance in noisy datasets and is appropriate for high variance models. Decision trees are able to manage significant disparities. One type of decision tree used in this study is REPTree also known as Classification and Regression Tree or CART.

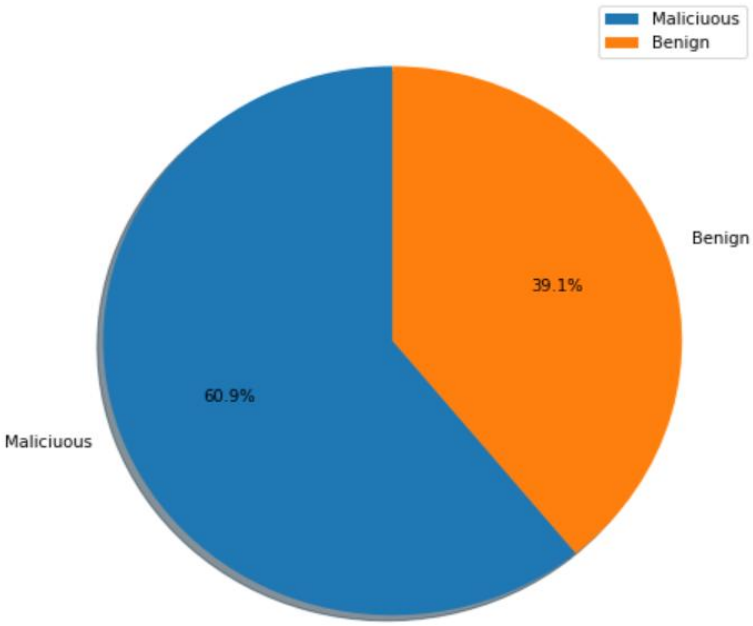
The best partition point is chosen at each stage of the tree construction when greedy algorithms are employed, which is a disadvantage of the bagged decision tree. Consequently, the trees at the end have nearly the same appearance. As a result, the disparity between each bag's prediction gets smaller. As a result, the expected model performs worse. The random forest ensemble algorithm can help get around this restriction. This happens when greedy partitioning algorithms are interfered with during the tree-building process by random forest ensemble methods. Only a random subset of the input attributes can be used to choose the partition points. The algorithm's performance is enhanced by decreasing the similarity between the bagged trees.

Therefore, even though it is assumed that the random forest ensemble approach overcomes the shortcomings of the bagging ensemble algorithm the current study compared the bagging and random forest ensemble algorithms for detecting DDoS attacks in SDN with the goal of determining which algorithm performs better.

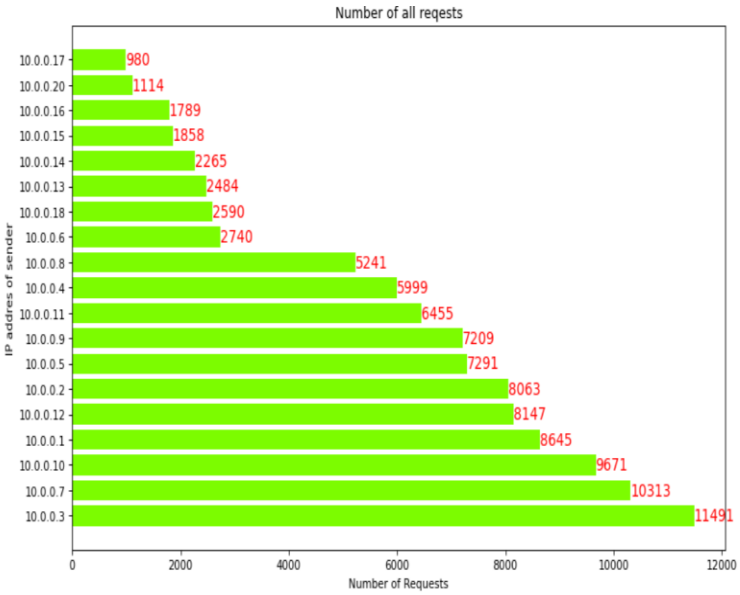
5. EXECUTION AND CONVERSATION

Analysis of Descriptive Data:

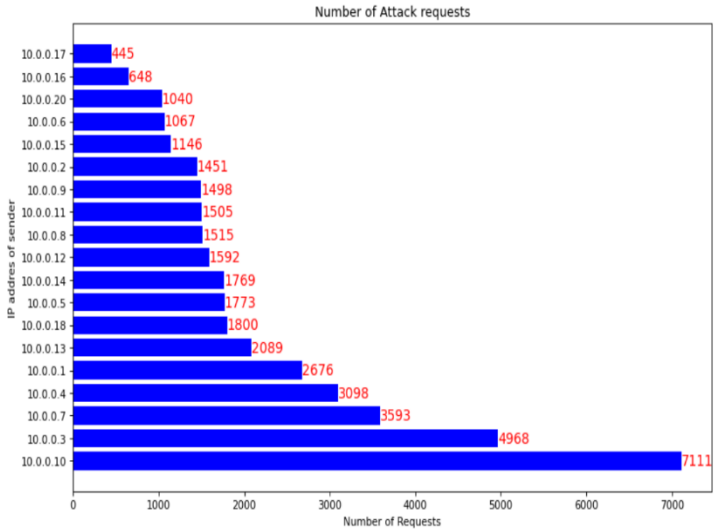
Figure 2 (A), (B) and (C) displays the descriptive analysis of the dataset that was used:


Maliciuous
Benign
39.1%
60.9%
Benign
Maliciuous

(A)


Number of all reqests
IP addres of sender
10.0.0.17
10.0.0.20
10.0.0.16
10.0.0.15
10.0.0.14
10.0.0.13
10.0.0.18
10.0.0.6
10.0.0.8
10.0.0.4
10.0.0.11
10.0.0.9
10.0.0.5
10.0.0.2
10.0.0.12
10.0.0.1
10.0.0.10
10.0.0.7
10.0.0.3
980
1114
1789
1858
2265
2484
2590
2740
5241
5999
6455
7209
7291
8063
8147
8645
9671
10313
11491
0
2000
4000
6000
8000
10000
12000
Number of Requests

(B)



(C)

Figure 2: Graphical Representation of the descriptive Dataset (A) Percentage of Malicious and Benign Requests (B) Number of all Requests (C) Number of Attack Requests in the dataset

Implementation –

The bagging ensemble approach has been used initially. Figure 3 below illustrates how the algorithms initial parameters were set.

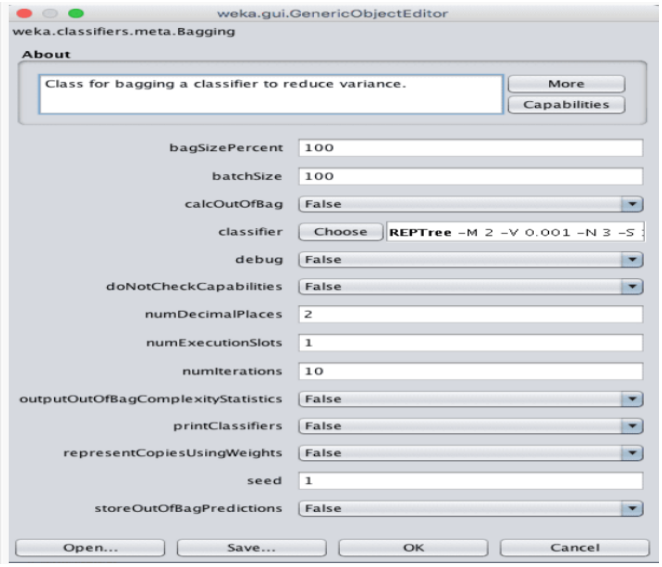


Figure 3: WEKA configuration for Bagging Algorithm

The following output is obtained with the above settings:

```
Test mode:    split 80.0% train, remainder test

=== Classifier model (full training set) ===

Bagging with 10 iterations and base learner

weka.classifiers.trees.REPTree -M 2 -V 0.001 -N 3 -S 1 -L -1 -I 0.0

Time taken to build model: 11.59 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0.03 seconds

=== Summary ===

Correlation coefficient                  0.9994
Mean absolute error                      0.0012
Root mean squared error                  0.0168
Relative absolute error                  0.2461 %
Root relative squared error              3.4481 %
Total Number of Instances            20869
```

Figure 4: Classification Results for Bagging Algorithm

The accuracy of the bagging ensemble method is 99.94 % as shown in Figure 4. According to earlier research [20] the procedures relative absolute error which was found to be 0.24% indicates that the model performs well in the classification. A root relative class error (RRSE) value of 3.448% can also be used to identify a satisfactory model zero is the ideal value.

The random forest ensemble algorithm was then used. The algorithms initial parameters were established as illustrated in Figure 5 below:

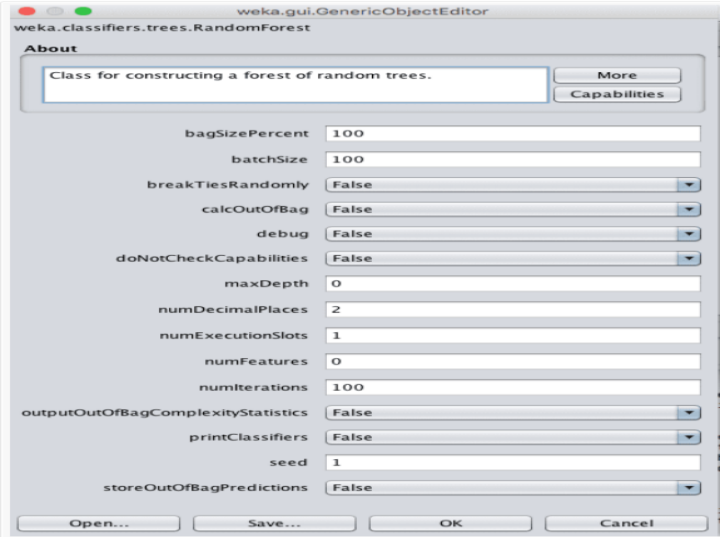


Figure 5: WEKA configuration for Random Forest Algorithm

With the above configuration, the following output has been achieved:

```
Test mode:    split 80.0% train, remainder test

=== Classifier model (full training set) ===

RandomForest

Bagging with 100 iterations and base learner

weka.classifiers.trees.RandomTree -K 0 -M 1.0 -V 0.001 -S 1 -depth 5 -do-not-check-capabilities

Time taken to build model: 23.43 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0.34 seconds

=== Summary ===

Correlation coefficient                  0.973
Mean absolute error                      0.0611
Root mean squared error                  0.1188
Relative absolute error                 12.8187 %
Root relative squared error             24.3228 %
Total Number of Instances           20869
```

Figure 6: Results of Classification for the Random Forest Algorithm

As seen in Figure 6 the Random Forest Ensemble algorithms accuracy rate seems to be 97.3%. Random Forest made 97.3% of the correct predictions whereas the Bagging Ensemble method made 99.94% of the correct predictions. This implies that in network discovery the bagging ensemble approach detects DDoS attacks more effectively than the random forest ensemble approach.

The model performs poorly for classification because the algorithms relative absolute error is greater than 1 or 12.81 % [14]. In a similar vein the observation reveals that the root relative square error is 24.32% suggesting the presence of another poorly fitting model. The bagging ensemble method worked well under these conditions (relative absolute error = 0.24% root relative square error = 3.448%).

6. RESULT AND DISCUSSION

Mean Absolute Error: The mean absolute error (MAE) shows the average size of errors in a set of estimations, regardless of whether the errors are positive or negative. In other words, by reporting the average difference between the estimated value and the actual observed value, it offers a simple indicator of forecasting accuracy. A lower MAE number indicates better model performance since the estimates are closer to the actual results. The formula for calculating MAE is given in Equation 1.

$$\mathrm{MAE} = \frac{1}{\mathrm{n}} \sum_{\mathrm{i}=1}^{\mathrm{n}} |\, \mathrm{Yi}\text{-}\overline{\mathrm{Yi}}| \quad \ldots\ldots\ldots\ldots (1)$$

Root Mean Squared Error: The root mean squared error (RMSE), which emphasizes larger discrepancies through its squared component, measures the average difference between estimated and actual values. It is widely used to assess the accuracy of a model; higher predictability is indicated by lower numbers. RMSE provides a more accurate representation of large errors than simple calculations. The formal computation is given by Equation 2.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Pi, A - Pi, M)^2} \ldots\ldots\ldots\ldots (2)$$

Random forest and Bagging are two ensemble learning techniques that are frequently used to address different machine learning challenges. The efficacy and accuracy of predictions made by tree-based algorithms can be improved by using bagging a straightforward and somewhat antiquated ensemble learning technique. Conversely random forest is a supervised machine learning method that can be applied to both classification and regression issues. The bootstrap sampling model has been refined. To obtain a genuine answer the random forest hypothesis proposes building several decision trees and combining them. Because random forests can minimize over-fitting they frequently offer higher accuracy levels than a single decision tree.

The Bagging Ensemble algorithm outperforms the Random Forest Ensemble technique in identifying DDoS attacks as demonstrated by the test results in the preceding sections. This can be explained by a number of factors including accuracy square root relative error (RRSE) and relative absolute error (RAE). Relative absolute error (RAE) is a metric used to assess a forecasting models efficacy. While values close to zero are ideal a low RAE value suggests that the bagging algorithm outperforms random forest.

The bagging ensemble algorithm employed in this investigation had an accuracy of 99.94%.

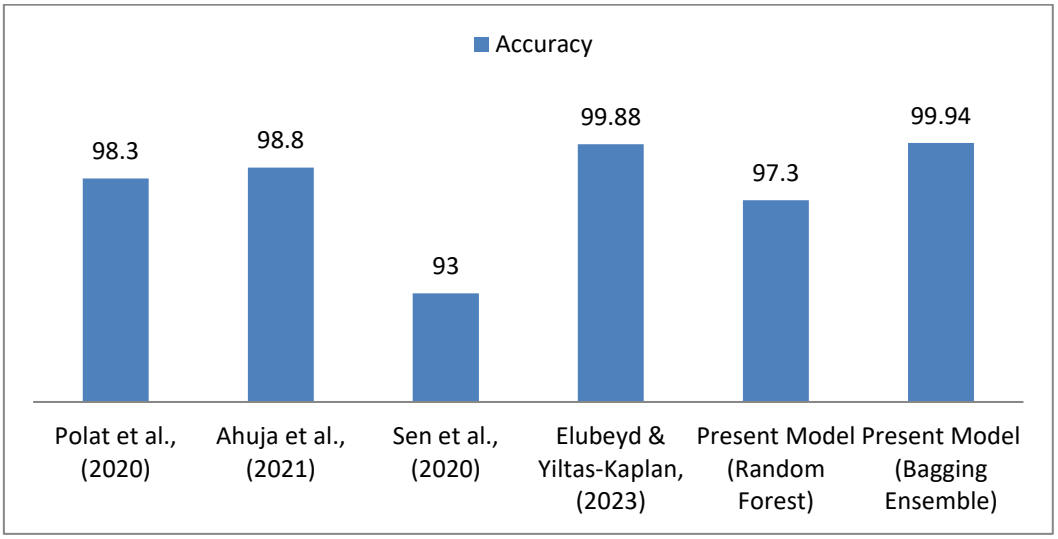


Figure 7: Comparative Analysis of Accuracies

The accuracy attained by the models taken into consideration in this study is contrasted with the accuracy attained by multiple earlier studies on machine learning-based DDoS attack detection in SDN in Figure 7. It is evident that the bagging ensemble model has shown the highest degree of accuracy when compared to all previous research.

Table 1 summarizes the experiment's results and contrasts the efficacy of bagging and random forest algorithms. It is simpler to evaluate, directly compare, and understand the relative benefits and drawbacks of various approaches thanks to this methodical presentation.

Table 1 (A) & (B): Examined Result

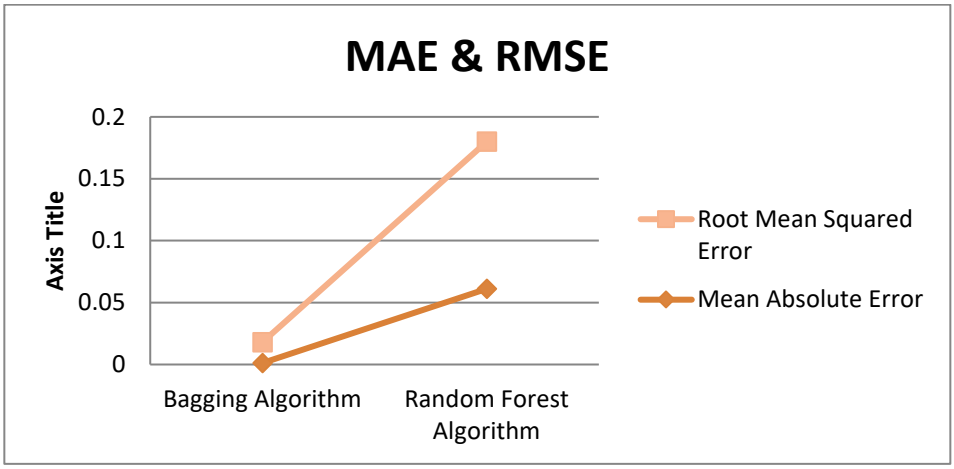
Testing Parameter Result	Bagging Algorithm	Random Forest Algorithm
Correlation Coefficient	99.94%	97.30%
Relative Absolute Error	0.2461%	12.8187%
Root Relative Squared Error	3.4481%	24.3228%

(A)

Testing Parameter Result	Bagging Algorithm	Random Forest Algorithm
Mean Absolute Error	0.0012	0.0611
Root Mean Squared Error	0.0168	0.1188

(B)

(A)



(B)

Figure 8 (A) & (B): Algorithm Wise Testing Outcome Dashboard

Figure 8 demonstrate that the bagging approach performs better in testing than random forest. Two machine learning algorithms are examined and contrasted in this article. Estimators are taught using the same fundamental methods.

Furthermore a real-time training dataset for future SDN-based private cloud research will be made available by this study. Additional machine learning methods will be assessed on this dataset in an attempt to further narrow the range of potential results and boost accuracy. Without a doubt this research has created new opportunities for future bagging evidence to be more accurate. This will facilitate future research by enabling low-error pruning bagging and bagging with classifiers other than REPTree.

7. CONCLUSION

The study examined the effectiveness of two ensemble learning techniques, bagging and random forest, to detect distributed denial-of-service (DDoS) attacks in a software-defined network (SDN). Even though both methods are frequently used to solve machine learning issues, the results clearly show that the bagging ensemble algorithm outperforms random forests in terms of accuracy and error reduction. Metrics like relative squared error (RRSE) and relative absolute error (RAE) show that bagging performs better; the RAE value is close to zero and the overall detection accuracy is 99.94%. These results show that bagging outperforms Random Forest in SDN-based DDoS detection by reducing variance and improving prediction accuracy. The findings demonstrate how important ensemble learning is to strengthening SDN defence against one of the most persistent cyber threats. This study develops machine learning-based security systems that can use bagging to adapt to changing attack tactics. Future improvement projects will primarily concentrate on producing real-time training datasets, particularly for SDN-based private cloud systems. Other machine learning methods besides REPTree and bagging will also be evaluated in order to improve detection

accuracy and reduce error margins. By opening up new channels for the integration of hybrid approaches, low-error pruning strategies, and other classifiers, this opens the door to more dependable and scalable cyber security solutions.

Acknowledgement: The authors thank their family and co-workers for their unwavering support.

Data Availability Statement: Resources in the public domain are the source of the data.

Funding Statement: Throughout its entire development this research study did not receive any funding.

Conflicts of Interest: The authors declare that they have no conflict of interest.

Ethical Approval: None of the writers of this article have conducted any research on humans or animals.

References

1. Benzekki K, El Fergougui A, El Belrhiti El Alaoui A. Software-defined networking (SDN): a survey. Secur Commun Netw. 2017;9. doi:10.1002/sec.1737.
2. Alashhab AA, Zahid MSM, Azim MA, Daha MY, Isyaku B, Ali S. A survey of low rate DDoS detection techniques based on machine learning in software-defined networks. Symmetry. 2022;14(8). doi:10.3390/sym14081563.
3. Cui J, Zhang Y, Cai Z, Liu A, Li Y. Securing display path for security-sensitive applications on mobile devices. Comput Mater Continua. 2018;55(1):17.
4. Pimpalkar AS, Patil AB. Detection and defense mechanisms against DDoS attacks: a review. In: 2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS). Piscataway (NJ): IEEE; 2015 Mar. p.1-6.
5. Zeb K, Baig O, Asif MK. DDoS attacks and countermeasures in cyberspace. In: 2015 2nd World Symposium on Web Applications and Networking (WSWAN). Piscataway (NJ): IEEE; 2015 Mar. p.1-6.
6. Cui Q, Zhou Z, Yuan C, Sun X, Wu QJ. Fast American Sign Language image recognition using CNNs with fine-tuning. J Internet Technol. 2018;19(7):2207-2214.
7. Bhayo J, Hameed S, Shah SA, Nasir J, Ahmed A, Draheim D. A novel DDoS attack detection framework for software-defined IoT (SD-IoT) networks using machine learning. SSRN Electron J. 2022. doi:10.2139/ssrn.4022910.
8. Alzahrani AO, Alenazi MJF. Designing a network intrusion detection system based on machine learning for software defined networks. Futur Internet. 2021;13(5). doi:10.3390/fi13050111.
9. Unknown. Cloudflare's 2025 Q3 DDoS threat report -- including Aisuru, the apex of botnets. The Cloudflare Blog [Internet]. 2025; https://blog.cloudflare.com/ddos-threat-report-2025-q3/.
10. Sahu D. https://www.facebook.com/VARINDIAMagazine. DDoS attacks more than doubled in 2025. https://www.varindia.com/news/ddos-attacks-more-than-doubled-in-2025-hitting-47-1-million-cloudflare-says.
11. Scott-Hayward S, O'Callaghan G, Sezer S. SDN security: a survey. In: 2013 IEEE SDN for Future Networks and Services (SDN4FNS). Piscataway (NJ): IEEE; 2013 Nov. p.1-7.
12. Alanazi F, Jambi K, Eassa F, Khemakhem M, Basuhail A, Alsubhi K. Ensemble deep learning models for mitigating DDoS attack in software-defined network. Intell Autom Soft Comput. 2022. doi:10.32604/iasc.2022.024668.
13. Ahuja N, Singal G, Mukhopadhyay D, Kumar N. Automated DDoS attack detection in software defined networking. J Netw Comput Appl. 2021;187. doi:10.1016/j.jnca.2021.103108.
14. Tonkal O, Polat H, Başaran E, Cömert Z, Kocaoğlu R. Machine learning approach equipped with neighborhood component analysis for DDoS attack detection in software-defined networking. Electronics. 2021;10(11). doi:10.3390/electronics10111227.
15. Srinivas SC, Avadhani PS, Roja PP. Machine learning based ensemble technique for DDoS attack detection in software-defined networking. Int J Comput Appl. 2023 May;185(6):22-25. doi:10.5120/ijca2023922712.
16. Elubeyd H, Yiltas-Kaplan D. Hybrid deep learning approach for automatic DoS/DDoS attacks detection in software-defined networks. Appl Sci. 2023;13(6). doi:10.3390/app13063828.
17. Ali TE, Chong YW, Manickam S, Yusoff MN, Yau KLA, Zoltan AD. A stacking ensemble model with enhanced feature selection for distributed denial-of-service detection in software-defined networks. Eng Technol Appl Sci Res. 2025 Feb;15(1):19232-19245.
18. Makuvaza A, Jat DS, Gamundani AM. Deep neural network (DNN) solution for real-time detection of distributed denial of service (DDoS) attacks in software defined networks (SDNs). SN Comput Sci. 2021;2(2). doi:10.1007/s42979-021-00467-1.
19. Alwabisi S, Ouni R, Saleem K. Using machine learning and software-defined networking to detect and mitigate DDoS attacks in fiber-optic networks. Electronics. 2022;11(23). doi:10.3390/electronics11234065.
20. Cichosz P. Book review: John Wiley & Sons, Chichester, 2015. ISBN 978-1-118-33258-0. 683 pp. USD 64.99. J Stat Softw. 2015 Aug;66(Book Review 2). Available from: http://www.jstatsoft.org/; http://www.wiley.com/WileyCDA/WileyTitle/productCd-111833258X.html.