OPEN ACCESS

ISSN: 2996-4954

Sandeep Kumar C, et al. (2022)100007
DOI: 10.71010/2996-4954/JCETAI/101

Research Article

Journal of Contemporary Education Theory & Artificial Intelligence

Efficient Machine Learning Approaches for Intrusion Identification of DDoS Attacks in Cloud Networks

Sandeep Kumar Chundru[1*], Srikanth Reddy Vangala[2], Ram Mohan Polam[3], Bhavana Kamarthapu[4], Ajay Babu Kakani[5], Sri Krishna Kireeti Nandiraju[6]

[1]*University of Central Missouri, Chundru.sandeepkumar@gmail.com*
[2]*University of Bridgeport, Srikanthreddy1043@gmail.com*
[3]*University of Illinois at Springfield, ramreddy.polam@gmail.com*
[4]*Fairleigh Dickinson University, kamarthapubhavana6@gmail.com*
[5]*Wright State University, kakani@outlook.com*
[6]*University of Illinois at Springfield, Srikrishna.nandiraju@gmail.com*

[*]**Correspondence:** Sandeep Kumar Chundru

Citation: Sandeep Kumar C, Srikanth Reddy V, Ram Mohan P, Bhavana K, Ajay Babu K, et al. (2022) Efficient Machine Learning Approaches for Intrusion Identification of DDoS Attacks in Cloud Networks. J Contemp Edu Theo Artific Intel: JCETAI/101.

Received Date: 05 November, 2022; **Accepted Date:** 16 November, 2022; **Published Date:** 23 November, 2022

Abstract

Recent trends have highlighted that DDoS contributes to most of the general attacks on networks. Networks struggle to distinguish between legitimate and malicious transmissions. Many reasons, encompassing the intricate, inflexible, costly, and vendor-specific architecture of modern networking devices and protocols, make it difficult to develop, test, and apply DDoS techniques. The BoT-IoT dataset, an extensive network traffic resource, is harnessed by the study to train and test a Decision Tree (DT) classifier that distinguishes DDoS attack patterns. The proposed methodology involves strenuous preparation steps for the data, including missing data imputation, feature mapping, one-hot encoding, and normalization, in order to ensure preprocessed data of high quality and consistent. The DT model performed trailer, reporting an accuracy, precision, recall, and F1-score of 99.9% across major points of evaluation. As revealed by comparative analysis to baseline models (GD and RF), the DT model outperforms for DDoS activity detection. The final results accentuate the robustness, interpretability and effectiveness of the Decision Tree classifier to be an interesting solution for real-time intrusion detection systems in the cloud environment.

Keywords: *DDoS Attack Detection, Machine Learning (ML), Decision Tree (DT), BoT-IoT Dataset, Cybersecurity.*

I. Introduction

Networks have now become the backbone of modern living, bearing exchange of communication, commerce, as well as the very necessary infrastructure. As a result, cybersecurity is a key aspect of how these networks are secured from a wide range of dangers. Some central cybersecurity technologies (anti-virus software, firewalls, and IDS) are deployed in order to defend against both internal attackers and external attackers [1]. Anti-virus software comes to the rescue while detecting and disarming malware and while filtering malicious traffic on the firewall, while IDSs keep a constant eye on network activities to detect suspicious patterns and possible breaches. These tools combine to provide a bottom-line defense, protecting data and systems from a world that gets more digitally centric with every passing day and ensuring the compromises of data and system availability, confidentiality, and integrity [2].

The increasing speed of worldwide communication has made cyberattacks, particularly One major worry is distributed denial of service assaults. During a DDoS assault [3], the attacks exploit numerous compromised network devices to massively overwhelm a target network or server, causing it to deny service to legitimate users [4]. The development of these attacks and the capacity to employ IoT botnets has ensured that such attacks cannot be blocked or mitigated. One of those is the 2016 DDoS attack on the Dyn DNS server, which knocked out the system but left the world connected [5]. As DDoS attacks continue to rise in frequency and complexity, an acute threat that requires better detection and mitigation mechanisms to withstand them emerges.

Traditional IDS normally uses signature-based techniques as one of their defensive strategies to detect known attacks by comparing traffic patterns to established attack signatures [6]. This way is useful for existing threats but lacks for new [7], complex attackers, such as DDoS. Also, traditional IDS are vulnerable to a high frequency of false positives (error of reporting legitimate traffic as an attack) and false negatives (the failure to detect a real attack) [8]. As cyber-attacks grow technologically more advanced and pervasive, it means that traditional security measures in the form of firewalls and access controls are insufficient to manage mass operation actions such as DDoS, thereby making the need for more adaptive and scalable intrusion detection systems.

Cloud computing has revolutionized the way businesses operate and has now introduced scalable, cost-saving solutions. However, the distributed nature of the cloud networks makes them particularly susceptible to DDoS attacks. To counter these threats, ML techniques are increasingly being employed to immediately identify and stop DDoS assaults by identifying abnormal traffic patterns and adapting to evolving attack vectors [9][10], which can flood all cloud nodes and create huge service outages. Such disruptions can be not only detrimental to cloud-hosted applications but also to services that depend on the cloud [11]. With the rapid shift of businesses to cloud environments, the demand for solid security measures to prevent DDoS attacks becomes more urgent. Common defense mechanisms such as firewalls and access control mechanisms have proved less adequate due to the influence of these attacks, and therefore [12]. To combat these attacks, it is also essential to have a sophisticated IDS that can identify and eliminate the resulting risks instantly.

A. Significance and Contribution

This study's importance stems from its ability to improve DDoS detection systems, which are crucial for safeguarding network infrastructures. The proposed study uses a robust dataset such as BoT-IoT, which allows for training and testing ML models with a strong framework that can provide better identification of attack patterns. The methodology, by involves systematic data pre-processing and the use of a tried and tested DT classifier, takes care that the results are built on quality, consistent data. Further strengthening the relevance of the study, the standard evaluation metrics further enhance the study’s validity, as there is a tangible evaluation of the model’s output. This study advances the state-of-the-art in DDoS tactics while being presented as a useful benchmark for future research in network security. The contribution of the study is given below:

- The paper uses the BoT-IoT dataset for DDoS attacks detection, using a fairly large amount of network traffic presented in the dataset, and therefore provides useful insights into the discipline of cybersecurity.
- The study uses strong pre-processing tools such as missing value imputation, feature mapping, one-hot encoding, and normalization to prepare high-quality data consistent enough to work with ML models.
- The paper has employed a Decision Tree classifier in detecting DDoS attack patterns, proving the classifier’s relevance and effectiveness in cybersecurity applications.
- The study uses various standard performance measures, including to evaluate the model's performance thoroughly, use the F1score, recall, accuracy, and precision produce an exhaustive report on its efficacy.

B. Justification and Novelty

This study justifies its approach by highlighting the acute necessity for DDoS detection technique with high robustness and working efficiency in the field of network security, a necessity that is met through approaches at the ML level. The BoT-IoT dataset, selected to reflect the patterns of traffic in networks, gives a worthy base for testing models of detection of attacks. The pre-processing processes, such as missing data imputation, feature mapping, and data normalization for the above data set, are necessary to maintain the data integrity and before modelling, to make sure the data is appropriate. Utilizing the DT classifier, one of the most tried and tested machine learning tools, this work seeks to accurately and effectively detect attack patterns. Common performance indicators include as accuracy, precision, recall, and F1-score are essential in guaranteeing thorough evaluation of the model and as well, it can be compared with other approaches in the field, which helps to provide some useful information in the ongoing study on DDoS detection.

C. Structure of the Paper

This study is organized as follows: Section II surveys related work on intrusion detection against DDoS attacks in cloud environments. Section III explains the methodology and materials. Section IV reports the experimental results. Section V concludes with a summary of findings.

II. Literature Review

The study on cloud network intrusion detection of DDoS assaults is summarized in this section.

Wani et al. (2019) A fresh dataset was created utilizing an intrusion detection system, and work was done on their own cloud environment with an attacking tool called Tor Hammer. This study uses a variety of ML methods! 99.7%, 97.6%, and 98.0% of the classification and overall accuracy were achieved using SVM, RF, and NB, respectively [12].

Chen et al. (2018) explain how to use XGBoost as a detection technique in SDN-based clouds. Additionally, employ the Mininet to build the SDN topology, with the POX serving as the SDN manager, and the attack tool Hyenae to simulate a real-world DDoS attack situation. In order to compete with other classifiers, Tcp Dump has collected a flow packet data set that the XGBoost classifier uses for DDoS detection. The results of the detection validate the rapidity of their method, scalability, reduced false positive rate, and increased accuracy [13].

Sangodoyin et al. (2018) evaluate the server's vulnerability to TCP-ACK, SYN, and Slow Loris attacks on the layer of the data plane. Their evaluation of DDoS attack detection is based on simulations run on Mininet. Disturbance from the confidence interval derived from the normal distribution of throughput polled without server attack forms the basis of their detection approach. Their simulation result demonstrates that flooding assaults may be identified with 99% accuracy when a window size of one minute is used [14].

Jiao et al. (2017) Analyze the suggested method using both simulated and actual datasets, such as the dataset from the Baidu Cloud Computing Platform, the datasheet for the 2007 CAIDA DDoS Attack, and the ISCX IDS dataset. According to experimental data, the suggested method can achieve a false alarm rate below 1% and an assault detection rate over 99% [15].

He, Zhang, and Lee (2017) declared that the solution utilizes statistical data from the virtual machines and the hypervisor of the cloud server to block the transmission of network packages to the external network. They thoroughly compare the performance of nine ML methods. According to their testing findings, there is a 99.7 percent success rate in identifying the four distinct kinds of DOS attacks [16].

Table I presents a literature review on DDoS attacks in cloud networks, summarizing the methodologies used, datasets employed, key findings, identified limitations, and proposed directions for future research.

Table I: Summary of Literature Review based on Intrusion Detection of DDoS Attacks in Cloud Networks.

Study	Methodology	Dataset	Key Findings	Limitations	Future Work
Wani et al. (2019)	Implemented ML algorithms (SVM, Naive Bayes, Random Forest) on a self-generated dataset using a cloud setup and Tor Hammer for DDoS attack simulation.	Custom dataset generated in own cloud using Tor Hammer.	Achieved accuracy: SVM – 99.7%, RF – 97.6%, NB – 98.0%.	Limited to a single attack tool and environment.	Expansion to more diverse attacks and real-world datasets.
Chen et al. (2018)	Used XGBoost classifier in SDN-based cloud setup with POX controller and Mininet, attack simulated via Hyenae, data captured using TcpDump.	Flow packet dataset captured via TcpDump.	High detection accuracy, low false positives, and high scalability.	May require further validation in large-scale production environments.	Test performance on more complex, real-time traffic in production-scale SDN networks.
Sangodoyin et al. (2018)	Simulated TCP-ACK, SYN, and Slowloris DDoS attacks in Mininet, detecting anomalies via deviation from normal throughput using a statistical confidence interval approach.	Traffic data from Mininet simulations.	Detected flooding attacks with 99% accuracy using a 1-minute window.	Focuses on only three types of DDoS attacks.	Extend detection mechanism to other DDoS variants and optimize response time.
Jiao et al. (2017)	Combined simulated and real datasets; tested detection approach using multiple benchmark datasets including ISCX, CAIDA, and Baidu cloud datasets.	ISCX IDS, CAIDA DDoS 2007, Baidu Cloud Computing datasets.	Detection rate >99%, false alarm rate <1%.	Limited discussion on real-time deployment and scalability.	Real-time implementation and evaluation in live cloud environments.
He, Zhang and Lee (2017)	Leveraged statistical data from hypervisor and VMs, applied nine ML algorithms, and compared performance in detecting outbound DDoS traffic.	Internal cloud server data from hypervisors and VMs.	identified more than 99.7% of the four DoS attack types.	Evaluation limited to specific ML models and predefined attack types.	Broaden scope to adaptive models and hybrid detection systems for emerging DDoS variants.

III. Methodology

The methodology for this study begins with the utilization of the BoT-IoT dataset, sourced from Kaggle, which contains comprehensive network traffic data suitable for DDoS detection research. The raw dataset undergoes a series of pre-processing steps, including missing data imputation, feature mapping, one-hot encoding, and normalization to guarantee the consistency and quality of the data. To facilitate efficient model assessment, after initial preparation, the dataset is divided into three parts: training, validation, and testing. On top of that, the processed data is run through a DT classifier to identify patterns of attacks. Properly, the model's efficacy is evaluated by displaying the outcomes of standard measures like as F1-score, recall, accuracy, and precision. The suggested methodology's process is depicted in Figure 1, detailing each step from dataset utilization to final result evaluation.

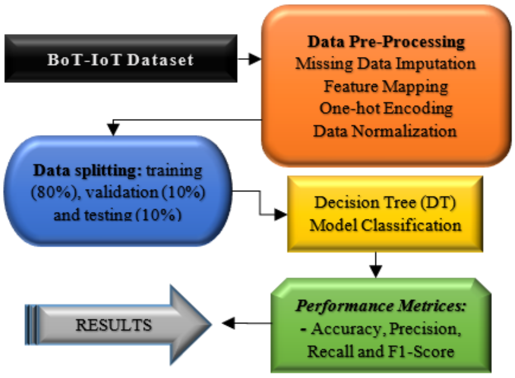
BoT-IoT Dataset
Data Pre-Processing
Missing Data Imputation
Feature Mapping
One-hot Encoding
Data Normalization
Data splitting: training (80%), validation (10%) and testing (10%)
Decision Tree (DT) Model Classification
Performance Metrices:
- Accuracy, Precision, Recall and F1-Score
RESULTS

Figure 1: Flowchart for DDoS Attacks in Cloud Networks.

The following is a quick discussion of the procedures in the recommended method:

A. Data Collection

This study makes use of the BoT-IoT dataset, which is accessible on Kaggle and was produced in the UNSW Canberra Cyber Range Lab's realistic network environment. The information, which comes in a variety of forms such as CSV files, Argus flow records, and actual packet capture (pcap) files, is intended for DDoS attack detection studies. Over 72 million records are included in fully compressed pcap files of 69.3 GB, while the extracted flow traffic measuring 16.7 GB is in a CSV format, offering a comprehensive resource for evaluating intrusion detection models.

Figure 2 illustrates the supply of samples between the two classes: Normal and Attack. It is clearly visible that the dataset is imbalanced in a way that the attack samples are largely higher than the number of normal samples. In particular, it has more than 25000 attack cases, and the number of normal traffic samples is less, around 8000. This imbalance draws more attention to the need for using strong models indicating identify DDoS attacks in a distorted data environment on cloud networks.

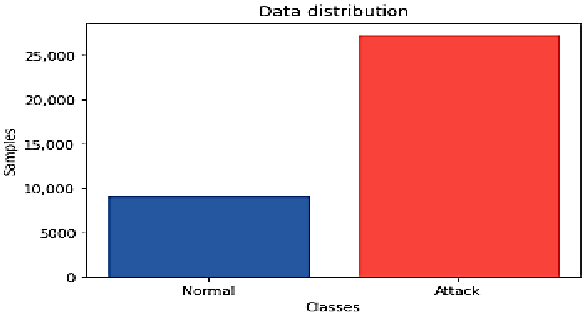

Figure 2: Data Distribution Plot for Binary Classification.

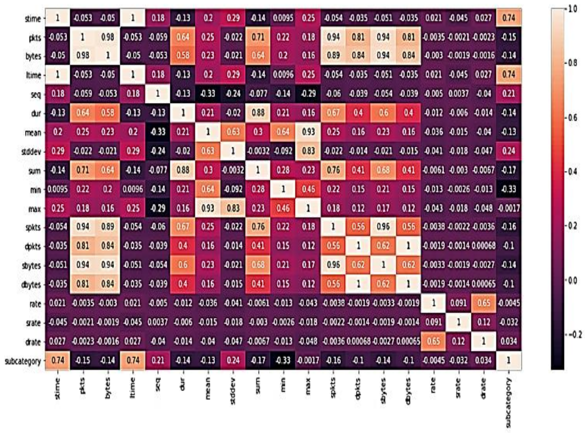

Figure 3: Correlation Matrix of Features.

Figure 3, the correlation between the dataset's various properties is depicted in the heatmap. With values ranging from -1 to 1, each cell displays the Pearson correlation between two attributes. Darker hues indicate weaker or negative connections, whereas lighter hues indicate greater positive correlations. It can be observed that some features show a high degree of correlation with each other, suggesting potential redundancy, while others are more independent. Understanding these relationships is crucial for feature selection, dimensionality reduction, and enhancing model performance during intrusion detection tasks.

B. *Data preprocessing*

The preprocessing of DDoS attack data in cloud environments using the BoT-IoT dataset is a critical focus. Essential steps in this process include missing data imputation, Feature Mapping, One-hot Encoding, and normalization. These techniques are grounded in well-established methodologies from existing literature, aiming to optimize model performance and improve the generalizability of the detection system across diverse network conditions. The following steps of preprocessing are given below:

- **Missing Data Imputation:** Missing data imputation and searching for missing data (both datasets had no missing data) [17].
- **Feature Mapping:** Machines are only able to recognised numerical values, yet the characteristics that are derived from network data comprise both numerical and symbolic elements. As a result, it is necessary to first translate each symbol characteristic into numerical values [18].
- **One-hot Encoding:** In intrusion detection tests, the majority of data are categorical in nature and may or may not be numerical. This was accomplished by using "One-hot encoding," which transforms each non-numeric value into a unique binary variable [19].
- **Data Normalization:** In order to prevent bias in favor of characteristics with higher values, data normalization is done. It involves adjusting each attribute's value to fall within a range that is proportionate. Each characteristic in each record falls within the same range of [0–1] and is normalised by the corresponding maximum value [20].

C. *Data Splitting*

The dataset partitioning was used for all ML and DL models in the DDoS assault investigation in cloud networks was 10% for testing, 80% for training, and 10% for validation.

D. *Classification of Decision Tree Model*

One supervised ML technique that is known to work effectively with proper configuration is the DT. Large data sets may be handled by it, and it handles both category and numerical data. A DT is made up of leaf nodes, which provide the expected outcomes, and the initial split is carried out by a root node. A DT can be represented using edges and nodes. Splitting is often determined by information gain and entropy [21].

In order to construct a DT, it is essential to determine the best way to split samples according to a feature and determine the weight values of the leaf nodes, which is accomplished by calculating Equations (1) and (2):

$$L_{split} = \frac{1}{2}\left[\frac{\left(\sum_{i\in I_L} g_i\right)^2}{\sum_{i\in I_l} h_L+\lambda} + \frac{\left(\sum_{i\in I_R} g_i)\right)^2}{\sum_{i\in h_R} h_L+\lambda} - \frac{\left(\sum_{i\in I} g_i\right)^2}{\sum_{i\in i} h_L+\lambda}\right] (1)$$

$$\omega_l^{(t)} = -\frac{\sum_{i\in I_l} g_i}{\sum_{i\in I_l} h_i+\lambda} \qquad (2)$$

As it can be shown, the first-order and second-order gradients and sample order are the only factors that affect the computation of this split score.

E. *Evaluation Metrics*

The standard classification measures, comprising F1-score, confusion matrix, recall, accuracy, and precision, are used to assess how well DL models identify DDoS assaults in cloud settings. It is common practice to create and display the difference between the predicted and actual classification results from a classification matrix using a confusion matrix. A confusion matrix, which shows the difference between the classifier's actual and predicted results, is commonly produced. To show how effectively a classification model works with test data where the true values are known, a table called a confusion matrix is often employed. These numbers, TP, TN, FP, and FN, are used to measure how the real model attempts to discriminate between attack (positive class) traffic & normal (negative class) traffic. It is critical to have an accurate interpretation of these metrics as a prerequisite for comparing model performance in intrusion detection. Below are discussed.

Accuracy: The calculation used is the method for calculating it, which is the ratio of all accurate projections to all forecasts made, or Equation (3):

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} (3)$$

Precision: It defines the number of true positives relative to all positives and is utilized to determine the amount of attack traffic that is truly identified accurately; the formula is calculated in Equation (4):

$$Precision = \frac{TP}{TP+FP} \qquad (4)$$

Recall: The attack rate is calculated as a proportion of total attack traffic, and recall indicates its ability to catch all real positives. Recall also determines the percentage of correctly anticipated actual positives. The formula above is obtained in Equation (5):

$$Recall = \frac{TP}{TP+FN} \quad (5)$$

F1 Score: This statistic provides a reasonable evaluation of the model's effectiveness in striking a balance between recall and accuracy. It is calculated as the midway figure between after accuracy and recall have been balanced, it is computed in Equation (6):

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (6)$$

These measurements are also used to evaluate how well various algorithms detect DDoS attacks.

IV. Result Analysis and Discussion

The experiment's findings from installing an ML model for cloud network intrusion detection of DDoS attacks are presented in this part. The experiments were performed on a system using a 3.3 GHz Intel dual-core i6 processor, 1 TB of RAM, and Windows 11 Pro. The models' performance was evaluated using four measures. Three metrics: F1-score, recall, and accuracy. Table II summarizes the results of the DT model's detection of distributed denial of service attacks.

Table II: Model performance of proposed Decision Tree Model.

Performance Metrics	**Decision Tree (%)**
Accuracy	99.9
Precision	99.9
Recall	99.9
F1-Score	99.9

The proposed DT model showed superb ability to identify DDoS assaults in cloud settings. The model was able to make effective differentiation in malicious versus normal traffic with 99.9% F1-score, recall, accuracy, and precision. Such findings identify the model’s resilience and fit to intrusion detection processes, making it an ideal candidate for deployment on a live cloud security system.

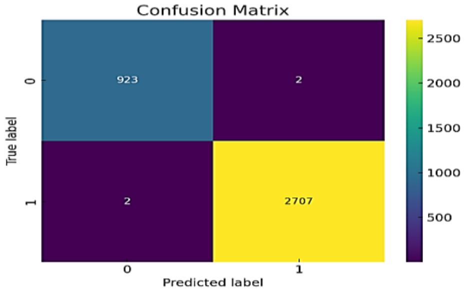

Figure 4: Confusion Matrix for DT Binary Classification.

The confusion matrix in Figure 4 displays the DT model's performance for DDoS attack detection. For all occurrences, 923 samples of normal traffic were correctly classified, whereas 2 were misclassified. In the same way, 2707 samples of DDoS attacks were properly detected, with only 2 of them being misclassified as traffic of normal traffic. These results point to the model’s superior classification capability, since errors are few, consistent with the high accuracy, precision, recall, and F1-score mentioned above.

A. Comparative Analysis and Discussion

The comparative analysis for intrusion detection and DDoS attack mitigation in cloud environments is discussed in the following subsection. ML, and DL models like GD, RF. Their Performance is compared with the help of key metrics such as accuracy, as seen by Table III.

Table III: Comparison analysis of proposed and baseline model based on DDoS Attack.

Model	**Accuracy (%)**
Gradient Descent [22]	97.7
Random Forest [23]	96
Decision Tree	99.9

In the comparison of model performance in Table III, the DT classifier, as the proposed model, outperforms both the GD and RF baseline models. The DT attained remarkable accuracy up to 99.9%, much higher than the GD model of 97.7% and the RF model of 96%. This tells that the DT model is better in identifying DDoS attack patterns in the dataset mentioned above, and hence the importance of completing the assignment with complexity. Although the GD and RF models give pretty good baseline results, the fact that the DT has higher accuracy indicates that this model is more appropriate to the problem, and provides a more reliable way to detect DDoS attacks can be found.

The proposed DT model has several benefits, such as outstanding accuracy (99.9%), making it very effective for detecting DDoS attacks. Decision-making processes are simple to see and comprehend due to its inherent interpretability. Thus, it is transparent. The model is able to deal well with non-linearities in the data, thus making it a good tool for the identification of complex attack patterns. Furthermore, DT requires very little data preprocessing and can handle both numerical as well as categorical data effortlessly. Using pruning techniques, the same model also becomes resistant to overfitting, which guarantees strong generalization to new data, making the proposed DDoS detection an effective and efficient option.

V. Conclusion and Future Work

This article demonstrates how well DT classifiers detect DDoS assaults in cloud networks. DDoS assaults are among the cloud's most intricate problems. The high model accuracy rate of 99.9% and outstanding performance among various evaluation metrics all prove the model’s soundness in identifying legitimate communications from malicious ones. Utilizing the BoT-IoT dataset in conjunction with comprehensive data pre-processing enhances the model's dependability, and therefore, it can be considered one of the possible solutions for real-time DDoS detection in cloud environments. Comparative analysis to baseline models, including GD and RF, underscores the DT model's superiority in handling the complexities of DDoS attack patterns.

Further work is suggested to investigate the possibility of applying more sophisticated ML methods, including ensemble models or DL based strategies, in order to examine how that could improve detection accuracy as well as scalability. Also, this model could be improved by dealing with the class imbalance in datasets while applying oversampling or data synthesis methods. Deployment in the real world in cloud infrastructures, as well as an ongoing adjustment to continually changing attack strategies, would yield meaningful information on the model's true-time effectiveness and ability to be adjusted.

References

1. H. Liu and B. Lang, "Machine learning and deep learning methods for intrusion detection systems: A survey," *Applied Sciences (Switzerland)*. 2019. doi: 10.3390/app9204396.
2. M. A. Khan, M. R. Karim, and Y. Kim, "A scalable and hybrid intrusion detection system based on the convolutional-LSTM network," *Symmetry (Basel).*, 2019, doi: 10.3390/sym11040583.
3. Z. Tan, A. Jamdagni, X. He, P. Nanda, and R. P. Liu, "A system for denial-of-service attack detection based on multivariate correlation analysis," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 2, pp. 447–456, 2014, doi: 10.1109/TPDS.2013.146.
4. A. Ain, M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Rank correlation for low-rate DDoS attack detection: An empirical evaluation," *Int. J. Netw. Secur.*, 2016.
5. P. Manso, J. Moura, and C. Serrão, "SDN-based intrusion detection system for early detection and mitigation of DDoS attacks," *Inf.*, 2019, doi: 10.3390/info10030106.
6. P. Mell and T. Grance, "The NIST definition of cloud computing," in *Cloud Computing and Government: Background, Benefits, Risks*, 2011. doi: 10.1016/b978-0-12-804018-8.15003-x.
7. V. Kolluri, "A Comprehensive Analysis on Explainable and Ethical Machine: Demystifying Advances in Artificial Intelligence," *TIJER - Int. Res. Journals*, vol. 2, no. 7, pp. 2349–9249, 2015.
8. M. Sookhak *et al.*, "Remote data auditing in cloud computing environments: A survey, taxonomy, and open issues," *ACM Computing Surveys*. 2015. doi: 10.1145/2764465.
9. J. Ruan, F. T. S. Chan, F. Zhu, X. Wang, and J. Yang, "A visualization review of cloud computing algorithms in the last decade," *Sustainability (Switzerland)*. 2016. doi: 10.3390/su8101008.
10. V. Kolluri, "A Pioneering Approach To Forensic Insights: Utilization AI for Cybersecurity Incident Investigations," *Int. J. Res. Anal. Rev.*, vol. 3, no. 3, 2016.
11. R. Chaâri *et al.*, "Cyber-physical systems clouds: A survey," *Comput. Networks*, 2016, doi: 10.1016/j.comnet.2016.08.017.
12. A. R. Wani, Q. P. Rana, U. Saxena, and N. Pandey, "Analysis and Detection of DDoS Attacks on Cloud Computing Environment using Machine Learning Techniques," in *Proceedings - 2019 Amity International Conference on Artificial Intelligence, AICAI 2019*, 2019. doi: 10.1109/AICAI.2019.8701238.
13. Z. Chen, F. Jiang, Y. Cheng, X. Gu, W. Liu, and J. Peng, "XGBoost Classifier for DDoS Attack Detection and Analysis in SDN-Based Cloud," in *Proceedings - 2018 IEEE International Conference on Big Data and Smart Computing, BigComp 2018*, 2018. doi: 10.1109/BigComp.2018.00044.
14. A. Sangodoyin, B. Modu, I. Awan, and J. Pagna Disso, "An Approach to Detecting Distributed Denial of Service Attacks in Software Defined Networks," in *Proceedings - 2018 IEEE 6th International Conference on Future Internet of Things and Cloud, FiCloud 2018*, 2018. doi: 10.1109/FiCloud.2018.00069.
15. J. Jiao *et al.*, "Detecting TCP-based DDoS attacks in Baidu cloud computing data centers," in *Proceedings of the IEEE Symposium on Reliable Distributed Systems*, 2017. doi: 10.1109/SRDS.2017.37.
16. Z. He, T. Zhang, and R. B. Lee, "Machine Learning Based DDoS Attack Detection from Source Side in Cloud," in *Proceedings - 4th IEEE International Conference on Cyber Security and Cloud Computing, CSCloud 2017 and 3rd IEEE International Conference of Scalable and Smart Cloud, SSC 2017*, 2017. doi: 10.1109/CSCloud.2017.58.
17. M. Padmanabhan, P. Yuan, G. Chada, and H. Van Nguyen, "Physician-friendly machine learning: A case study with cardiovascular disease risk prediction," *J. Clin. Med.*, 2019, doi: 10.3390/jcm8071050.
18. Y. Yang, K. Zheng, C. Wu, X. Niu, and Y. Yang, "Building an effective intrusion detection system using the modified density peak clustering algorithm and deep belief networks," *Appl. Sci.*, 2019, doi: 10.3390/app9020238.
19. A. U. H. Qureshi, H. Larijani, N. Mtetwa, A. Javed, and J. Ahmad, "RNN-ABC: A new swarm optimization based technique for anomaly detection," *Computers*, 2019, doi: 10.3390/computers8030059.
20. K. Siddique, Z. Akhtar, H. G. Lee, W. Kim, and Y. Kim, "Toward bulk synchronous parallel-based machine learning techniques for anomaly detection in high-speed big data networks," *Symmetry (Basel).*, 2017, doi: 10.3390/sym9090197.
21. M. Hafsa and F. Jemili, "Comparative study between big data analysis techniques in intrusion detection," *Big Data Cogn. Comput.*, 2019, doi: 10.3390/bdcc3010001.
22. O. Ali and P. Cotae, "Towards DoS/DDoS Attack Detection Using Artificial Neural Networks," in *2018 9th IEEE Annual Ubiquitous Computing, Electronics and Mobile Communication Conference, UEMCON 2018*, 2018. doi: 10.1109/UEMCON.2018.8796637.
23. N. Bindra and M. Sood, "Detecting DDoS Attacks Using Machine Learning Techniques and Contemporary Intrusion Detection Dataset," *Autom. Control Comput. Sci.*, vol. 53, no. 5, pp. 419–428, Sep. 2019, doi: 10.3103/S0146411619050043.
24. Chinta, P. C. R., & Karaka, L. M. (2020). Agentic AI and Reinforcement Learning: Towards More Autonomous and Adaptive AI Systems.
25. Kalla, D., Kuraku, D. S., & Samaah, F. (2021). Enhancing cyber security by predicting malwares using supervised machine learning models. *International Journal of Computing and Artificial Intelligence*, *2*(2), 55-62.
26. Katari, A., & Kalla, D. (2021). Cost Optimization in Cloud-Based Financial Data Lakes: Techniques and Case Studies. *ESP Journal of Engineering & Technology Advancements (ESP-JETA)*, *1*(1), 150-157.
27. Kalla, D., Smith, N., Samaah, F., & Polimetla, K. (2021). Facial Emotion and Sentiment Detection Using Convolutional Neural Network. *Indian Journal of Artificial Intelligence Research (INDJAIR)*, *1*(1), 1-13.
28. Routhu, K., Bodepudi, V., Jha, K. M., & Chinta, P. C. R. (2020). A Deep Learning Architectures for Enhancing Cyber Security Protocols in Big Data Integrated ERP Systems. *Available at SSRN 5102662*.

29. Kuraku, S., & Kalla, D. (2020). Emotet malware—a banking credentials stealer. *Iosr J. Comput. Eng*, *22*, 31-41.
30. Kalla, D., & Samiuddin, V. (2020). Chatbot for medical treatment using NLTK Lib. *IOSR J. Comput. Eng*, *22*, 12.
31. Routhu, K., & Jha, K. M. (2021). Leveraging Big Data Analytics and Machine Learning Techniques for Sentiment Analysis of Amazon Product Reviews in Business Insights. *Available at SSRN 5106490*.
32. Chinta, P. C. R., & Katnapally, N. (2021). Neural Network-Based Risk Assessment for Cybersecurity in Big Data-Oriented ERP Infrastructures. *Neural Network-Based Risk Assessment for Cybersecurity in Big Data-Oriented ERP Infrastructures*.
33. Karaka, L. M. (2021). Optimising Product Enhancements Strategic Approaches to Managing Complexity. *Available at SSRN 5147875*.
34. Boppana, S. B., Moore, C. S., Bodepudi, V., Jha, K. M., Maka, S. R., & Sadaram, G. AI and ML Applications in Big Data Analytics: Transforming ERP Security Models for Modern Enterprises.