International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(11s) 2026 pp. 816-826
https://doi.org/10.70917/ijcisim-2026-3817
Article

AN ADAPTIVE MACHINE LEARNING FRAMEWORK FOR REAL-TIME WEB APPLICATION ATTACK AND ANOMALY DETECTION

Vinod Jagannath Kadam[1], Nitin Vasant Ambhore[2], Shrikant Subhash Said[3], Zaid Salim Inamdar[4], Munir Bashir Sayyad[5]

[1] Dr. Babasaheb Ambedkar Technological University, Lonere.
Email: vjkadam@dbatu.ac.in
[2] Dr. Babasaheb Ambedkar Technological University, Lonere.
Email: nambhore@gmail.com
[3] Dr. Babasaheb Ambedkar Technological University, Lonere.
Email: shrikant.said@gmail.com
[4] Dr. Babasaheb Ambedkar Technological University, Lonere.
Email: zaidinamdar08@gmail.com
[5]Reliance JIO, Navi Mumbai.
Email: munirsayyad@gmail.com

Abstract: The project attempts to detect and prevent growing application-layer DDoS attacks, giving insights into attack patterns and tools for increased cybersecurity measures. The project concentrates on attacks at the HTTP level, and will dissect the techniques and tools to create a special approach to help understand and defend against the emerging cyber threats. The project needs to be oriented towards accessibility of tools immediately especially with the growing DDoS threats. This is critical for active defence against proliferation of attack tools. The project aims to provide network managers and cybersecurity professionals with the tools required to ensure the security of online services. Finally, it offers the users and organisations resilient protection against the new generations of DDoS attacks. We also used ensemble models (VC (RF, DT) and Stacking Classifier (RF, DT, LGBM)) to improve the performance. These enhancements aim to enhance the ability to identify incidents of cyberbullying.
Keywords: DDoS, DDoS tools, machine learning, deep learning.

1. INTRODUCTION

One of the most severe and complex security challenges to computer networks is DDoS attack [1] [2]. There was a considerable increase in application-layer attacks in Q1 2022, with HTTP-layer DDoS attacks increasing by 164% YoY and 135% QoQ. In terms of the industry, it was Consumer Electronics as the segment experienced the highest increase at 5,086% QoQ. Online Media was placed second with an increase of 2,131% in assaults QoQ, followed by Computer Software firms who ranked third with an increase of 76% QoQ and 1,472 YoY in attacks [2].

DDoS Attack is an onslaught of Internet traffic intended to disrupt the normal operation of a targeted server, service or network. In this attack, the attacker uses a network of computers or other devices (known as a ‘‘botnet’’) to send a huge amount of data to the target system, making it inaccessible to legitimate users [4]. The DDoS attacks are complex attacks due to 1) having a lot of traffic coming from a great number of sources, and 2) the traffic appears to be originating from a large number of sources [5].

There are several forms of DDoS attack, such as applicationlayer attacks. DDoS attacks at the application layer attack the application layer of the attacked system with the aim of consuming the resources of the attacked system or crashing the application system. This can be done by HTTP flooding and Slowloris attacks. The objective of such an application-layer DDoS attack is to overload the network in such a way that systems crash or become unavailable [6, 7, 8].

The availability of DDoS attack tools is a main factor of increase of DDoS attacks. They can be intentionally employed to flood servers and websites with traffic until they become unusable. These readily available tools can be bought from the dark web or simply downloaded from the internet and any individual with little or no technical experience can use them and carry out devastating attacks on DDoS [4], [9], [10], [11].

2. LITERATURE SURVEY

The solutions with IoT are entering into every field where people work, into their smart home, into their smart power grid, into smart automation etc. These devices extend the attack surface and were a prime target for the attacker, since the resource-constrained system doesn't provide a lot of options for implementing heavy security measures. In the less secure and often unsupervised case of IoT devices, the attacker can easily create a botnet army to start Denial of Service attack on large scale [1, 2, 17]. Therefore, this study [1] suggests a new Attack Detection approach using Machine Learning algorithm for CIoT to detect the attack traffic. This solution works on local attributes particular to IoT network to enable low-cost machine learning classifiers to identify attack, at the local router. The experimental results show that the proposed approach is able to achieve the highest accuracy of 0.99, thus demonstrating the robustness and reliability of the proposed approach in IoT network.

Today, the Internet is an emerging technology for remote control of industrial applications where one site needs to operate remotely another site (e.g. power plants controllers). DoS attacks can severely interrupt the Internet and so endanger the operation of such network-based control systems. The method of location concealing technology has been proposed for providing security to Internet application site over the overlay networks. This paper is a survey of many previous works on the subject [2]. In this research we study the architecture of an overlay network interface so that communication services among geographically scattered application sites are protected against DoS attacks. In [17, 18] a new architecture, called OPL, is proposed to proactively protect application sites from DoS attacks. Through simulation, the research proves that the probability of DoS attacks to interfere communications services in the OPL architecture is very low. 75% of communication channels are available even if 50% of attack nodes are attacked by Distributed DoS attack.

In the last several years, the DDoS attack has been most often selected by most hackers. It is attributed a number of and type of hardships it can bring. There's a large number of hackers and specialists in this particular field that have developed packages and programs that attack a network of a different sort using DDoS. However, it is crucial to assess and compare the power of the DDoS attack that is being sent through these technologies, and then create the proper defense against it. In this research [4], the performance of three DDoS attack tools are compared and analyzed on characteristics such as time to successfully start an attack, traffic rate and packet size. The following tools were chosen for evaluation: Slowloris, GoldenEye and Xerxes. The experimental findings show that Xerxes is better than other programs to launch a DDoS attack[21, 23].

SDN is a new paradigm that separates the forwarding hardware from the control decisions. It will likely greatly ease network management and enable innovation and evolution. SDN logically centralises network intelligence in software based controllers (the control plane) and the network devices (OpenFlow Switches) become basic packet forwarding devices (the data plane) that may be programmed via an open interface (OpenFlow protocol). Such separation between control plane and data plane presents some challenges including security, reliability, load balancing and traffic engineering. DoS and DDoS attack are terrible security issues in SDNs [5]. For SDNs, DoS/DDoS attacks can be launched to overload the control plane, the data plane, or the communication channel, for example. Attack to control plane can result in a failure of the entire network, while attack to data plane or communication channel can result in packet drop and the network will be unavailable. There are several contributions that shed light on the DoS/DDoS attack problem in SDNs in this study, which provide comprehensive coverage on the background of the problem, including attacks and analysis of the current solutions [19, 20]. Specifically, we review and categorize the most recent strategies used to deal with both DoS and DDoS attacks in SDNs, both from an intrinsic and an extrinsic point of view. Furthermore, the countermeasures are the same as their target, either detection, mitigation or prevention or an organised decline. Also, various methodologies and tools applied in implementing the modified solutions are discussed. Last, we identify some research directions that can be taken in the future to defend against a DoS/DDoS attack in an SDN [21].

The issue of DDoS is growing fast. The number and variety of the attacks and the defence approaches are incredible. In this work [6] two taxonomies are introduced which classify the assaults and defences, thereby giving the scholars a better understanding of the problem and the current solution space. The criteria for attacking have been defined to emphasize similarities and key features of the attacking methods which are fundamental to the problem and hence to the design of defence. The existing DDoS defences are categorized according to the design decisions of the defence taxonomy and the pros and cons of the proposed solutions are illustrated according to the categorized existing DDoS defences [4, 23].

3. METHODOLOGY

i) Proposed Work:

The proposed method for DDoS attack detection offers a broader solution rather than just detecting the patterns of DDoS attack [20]. Instead, a more global perspective is taken and popular attack tools are analysed in terms of their availability and effectiveness. This wider perspective helps to enhance our understanding of cyber security threats, and to build more effective defences. It's a system that's able to be responsive to the evolving threats, and not to be stuck searching for a single type of attack. To enhance the performance, we introduced the advanced ensemble models like VC which combines the models RF and DT and Stacking Classifier which has foundation learners RF and DT [20]. The ensemble approach attempts to improve the accuracy of cyberbullying detection. Additionally, a friendly Flask framework with SQLite was employed to secure signup and signin script which aids in the user testing and offers inputs and results. The enhancements add to the project's versatility, offering more model topologies for precise results and making interactions more efficient, ensuring the robustness and practicality of the project.

ii) System Architecture:

The system architecture starts with data preparation, by applying the datasets of NSL-KDD [13] and NBOT-IOT. Feature selection prepares data for efficient analysis, and is the next step. Then three classifiers are used: MLP with different optimisation algorithms (SGD, LBFGS, Adam), Extension stacking classifier and Extension voting classifier. Such classifiers cooperate to improve the prediction accuracy. This entire structure provides good analysis and prediction, so it is a flexible and effective solution to detect and resist the DDoS attacks in the network security.

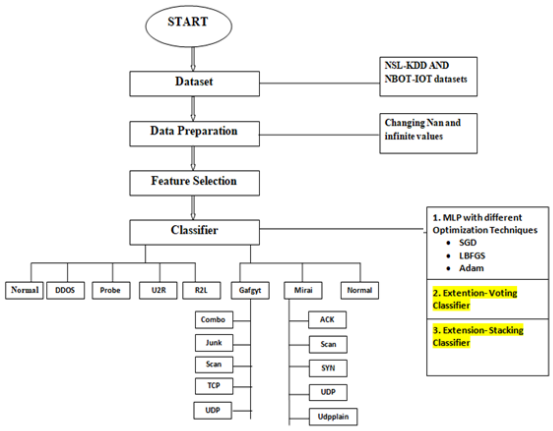

Fig 1 Proposed architecture

iii) Dataset collection:

NSL-KDD Dataset:

The NSL-KDD dataset is a benchmark dataset for evaluation of intrusion detection systems. The data used here is the original KDD Cup 99 dataset but with some modifications to overcome its limitations. NSL-KDD is an excellent dataset for cybersecurity training and testing machine learning models due to its rich nature of regular and different types of attacks in network traffic [13].

NBOT-IOT Dataset:

The NBOT-IOT dataset is for network behaviour analysis of the IoT. It comprises data generated by different IoT devices which provides insights about the communication patterns and possible risks in IoT networks. The data

presented in this dataset is critical for creating machine learning models that can identify anomalies and potential cyber threats in the IoT environment.

	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	...	dst_host_srv_count	dst_host_same_srv_rate	dst_host_d
0	0	tcp	http	SF	181	5450	0	0	0	0	...	9	1.0	
1	0	tcp	http	SF	239	486	0	0	0	0	...	19	1.0	
2	0	tcp	http	SF	235	1337	0	0	0	0	...	29	1.0	
3	0	tcp	http	SF	219	1337	0	0	0	0	...	39	1.0	
4	0	tcp	http	SF	217	2032	0	0	0	0	...	49	1.0	

5 rows × 42 columns

Fig 2 NSL KDD dataset

iv) Data Processing:

Data Processing is the transformation of raw data into information that will benefit a business. In general, data scientists work with data. This includes collecting, organizing, sanitizing, verifying, examining and converting data to create understandable representations (charts, reports etc.). The data processing may be carried out with three methods i.e. manual, mechanical and electronic. The notion is to add value to information and ease decision making. This enables businesses to optimize their workflow and make prompt decisions. It is here that automated solutions for processing data such as computer software development come in. It can be useful to transform vast amounts of data, such as big data, into valuable information for quality management and decision making.

v) Feature selection:

The task of feature selection is to determine the most consistent, non-redundant and relevant features to use for building models. In the era of increasing size and variety of datasets, it is of utmost importance to shrink the size of the datasets systematically. Feature selection is to improve the performance of prediction model and reduce the computational cost of modelling.

One of the primary steps in feature engineering is feature selection, which selects the most important features to pass to the machine learning algorithms. To decrease the number of input variables, redundant or irrelevant features are removed and the set of features are reduced to represent the most relevant characteristics to the machine learning model using feature selection strategies. The benefits of the feature selection process over letting the machine learning model pick the features.

vi) Algorithms:

In this project, we use MLP and use the SGD as the optimisation algorithm. It is a very effective algorithm to train the neural networks. SGD is a cheap version of the gradient descent algorithm that randomly selects a subset of training data or "batch" for each iteration. The combination of MLP and SGD is suitable for this research because it learns and adapts to complex patterns in the NSL-KDD and NBOT-IOT data sets, and can be used as a stable foundation for detecting DDoS attacks due to its ability to explore the high-dimensional feature space [20].

MLP - sgd

```
from sklearn.neural_network import MLPClassifier
# instantiate the model
clf = MLPClassifier(solver='sgd')

# fit the model
clf.fit(X_train, y_train)

#predicting the target value from the model for the samples

y_hat = clf.predict(X_test)
```

Fig 3 MLP-SGD

MLP COMBINED WITH Limited-memory Broyden-Fletcher-Goldfarb-Shanno (lbfgs): Quasi-Newton optimisation techniques are a class of optimisation techniques that aim to find the minimum of a function without computing its derivatives: lbfgs is a member of this family that maintains an approximation of the inverse Hessian matrix to update the model parameters. It works very well in the case that the dataset isn't very large and the model has moderate number of parameters. If the data set is moderate size and the model has a reasonable number of

parameters, then lbfgs may be appropriate for THIS project. This tends to be quicker than other optimisation strategies when the data can be stored conveniently in memory.

MLP - lbfgs

```
from sklearn.neural_network import MLPClassifier
# instantiate the model
clf = MLPClassifier(solver='lbfgs')

# fit the model
clf.fit(X_train, y_train)

#predicting the target value from the model for the samples

y_hat = clf.predict(X_test)
```

Fig 4 MLP- LBFGS

MLP - Adam: Adaptive Moment Estimation is an adaptive learning rate optimisation algorithm that is a combination of momentum and RMSprop. It updates each parameter with a different learning rate, depending on the previous gradients making it suitable for use if the gradients are sparse or there is noise. Adam is known for its efficiency, and is typically the standard optimiser used in many DL applications. It involves methods to control step sizes and to attenuate the previous gradients in an exponential manner [21].

MLP - adam

```
from sklearn.neural_network import MLPClassifier
# instantiate the model
clf = MLPClassifier(solver='adam')

# fit the model
clf.fit(X_train, y_train)

#predicting the target value from the model for the samples

y_hat = clf.predict(X_test)
```

Fig 5 MLP-Adam

The Stacking Classifier takes into account predictions of basic classifiers such as RF or DT, and then uses the final estimator (LGBMClassifier). It incorporates numerous classifiers to boost its accuracy and ability to withstand novel forms of DDoS attacks, suitable for the project's network security objectives.

Stacking Classifier

```
from sklearn.tree import DecisionTreeClassifier
from lightgbm import LGBMClassifier
from sklearn.ensemble import StackingClassifier

estimators = [('rf', forest),('dt', DecisionTreeClassifier(random_state=1))]

clf1 = StackingClassifier(estimators=estimators, final_estimator=LGBMClassifier(n_estimators=10))


clf1.fit(X_train,y_train)

y_pred = clf1.predict(X_test)
```

Fig 6 Stacking classifier

The Voting Classifier combines a GridSearchCV-optimized RF Classifier and a DT Classifier and the voting is done using a “soft” voting method. This ensemble approach utilizes the weighted averages of class probabilities to enhance the predicting performance, and is applicable to DDoS attack detection. Using diversity of classifiers can improve the defence against different cyber threats, leading to a more robust and effective security approach.

Voting CLassifier

```
from sklearn.ensemble import RandomForestClassifier, VotingClassifier, AdaBoostClassifier
from sklearn.tree import DecisionTreeClassifier

rfc = RandomForestClassifier()
parameters = {
    "n_estimators":[25],
    "max_depth":[20]

}

from sklearn.model_selection import GridSearchCV
forest = GridSearchCV(rfc,parameters,cv=10)

clf2 = DecisionTreeClassifier(random_state=100)

eclf1 = VotingClassifier(estimators=[('rf-parameter', forest), ('dt', clf2)], voting='soft')
eclf1.fit(X_train, y_train)
y_pred = eclf1.predict(X_test)
```

Fig 7 Voting classifier

4. EXPERIMENTAL RESULTS

Precision: The number of accurately classified instances or samples over the total number of classed as positives. Thus the formula to compute the precision is given by:

Precision = True positives/ (True positives + False positives) = TP/(TP + FP)

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

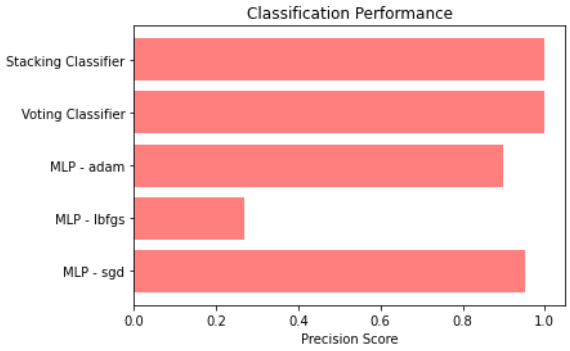

Fig 6 Precision comparison graph

Recall: Recall measures how well your model can find all relevant instances of a class. It is the ratio of the number of positive observations that it correctly predicts to the number of positive observations that are actually present that provides the understanding of the completeness of the model in capturing the positive class:

$$Recall = \frac{TP}{TP + FN}$$

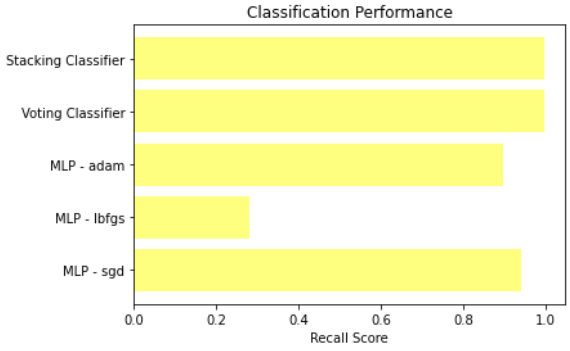

Fig 7 Recall comparison graph

Accuracy: In classification problem, it is the fraction of total predictions that are correct and is a measure of overall accuracy of a model's predictions.

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

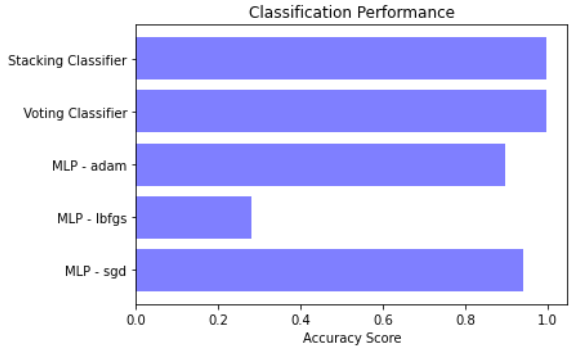

Fig 8 Accuracy graph

F1 Score: The F1 Score is the harmonic average of precision and recall and provides a balanced measurement between these two, and is a suitable index for imbalanced data.

$$\text{F1 Score} = 2 * \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} * 100$$

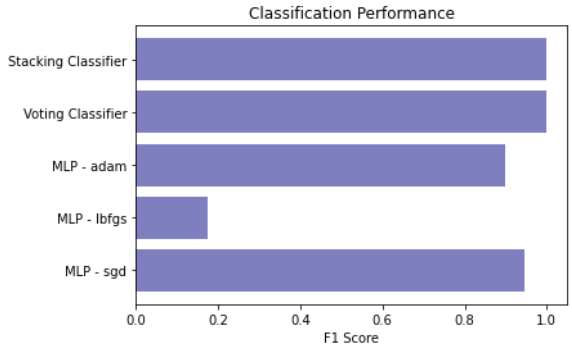

Fig 9 F1Score

	ML Model	Accuracy	f1_score	Recall	Precision
0	MLP - sgd	0.943	0.947	0.943	0.953
1	MLP - lbfgs	0.282	0.175	0.282	0.269
2	MLP - adam	0.898	0.898	0.898	0.899
3	Voting Classifier	0.998	0.998	0.998	0.998
4	Stacking Classifier	0.999	1.000	0.999	1.000

Fig 10 Performance Evaluation

Fig 11 Home page

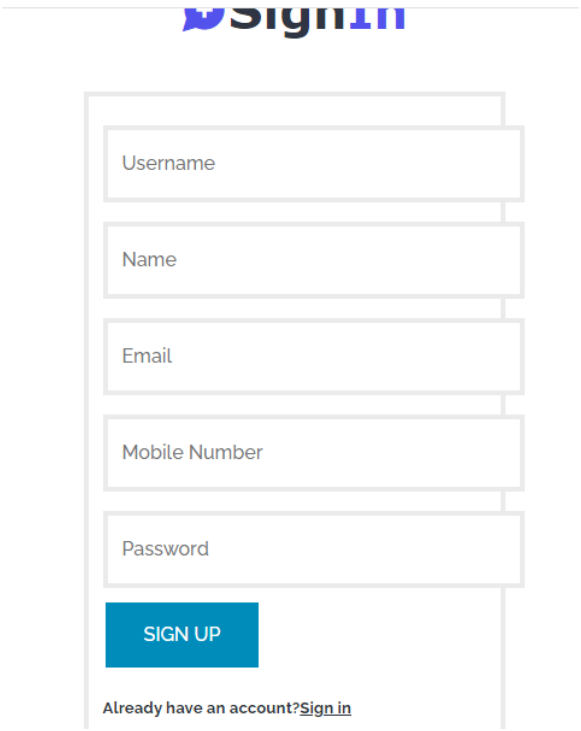

Fig 12 Signin page

Fig 13 Login page

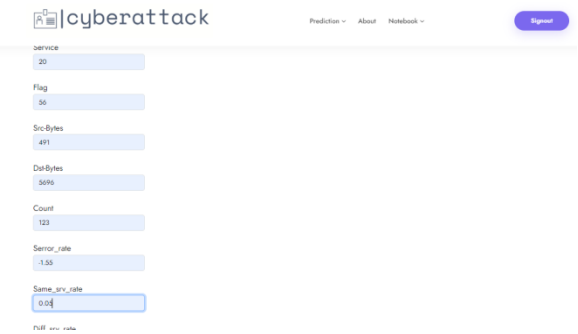

Fig 14 User input

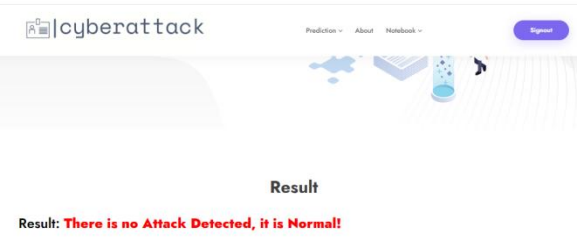

Fig 15 Predict result for given input

5. CONCLUSION

The research will be very valuable for the field of cybersecurity with innovative options for the detection and mitigation of DDoS attacks that will make computer networks more resilient [16, 17]. An in-depth analysis of different data sets (KDD-CUP and NBOT-IOT) offered significant information on network traffic attributes and probable attacks that offered a good foundation for building the efficiency of the models. A variety of optimisers such as as SGD, lbfgs and adam were tested to optimize the MLP [21] models, and they were found to be the most effective for detecting DDoS attack, thereby enhancing the robustness of cybersecurity defenses. And included voting and stacking classifiers. The combination of predictions of various models shows originality. This technique helps improve the accuracy and resilience of the forecasts to a large variety of types of cyber threats. Flask framework + SQLite for user signin / sign up with a user-friendly front-end is practical. The system is easy to use with simple input, predictions and interaction to enhance the usefulness in the real world.

6. FUTURE SCOPE

Going forward, the project can be expanded to include new developments in machine learning algorithms, which could enhance the accuracy and efficiency of detecting DDoS attacks.Continuous improvements in the machine learning algorithms can be integrated into the project to further refine the DDoS attack detection accuracy and efficiency [21]. The system could be enhanced by additional research and application of state-of-the-art algorithms and models to better adapt to evolving cyber threats. Development of real time detection techniques and reaction strategies for future scope of the research. It will be crucial to have technologies integrated that will allow an immediate detection and Mitigation of DDoS assaults, to reduce the effects of these threats on network resources [23]. The use of sophisticated behavioural analytic methods can help the future development of the project. Understanding the usual behaviour of networks and devices helps the system better spot anomalies associated with DDoS attacks, therefore helping a proactive approach to cybersecurity. The future scope of the project is scalability and adaptability as networks and cyber threats are always changing. To be effective in the ever-evolving field of cybersecurity, the system must be capable of handling a larger volume of data, various types of attacks and new technologies.

References

1. B. B. Gupta, P. Chaudhary, X. Chang, and N. Nedjah, ‘‘Smart defense against distributed denial of service attack in IoT networks using supervised learning classifiers,’’ Comput. Electr. Eng., vol. 98, Mar. 2022, Art. no. 107726.

2. H. Beitollahi and G. Deconinck, ‘‘An overlay protection layer against denial-of-service attacks,’’ in Proc. IEEE Int. Symp. Parallel Distrib. Process., Apr. 2008, pp. 1–8.
3. O. Yoachimik, ‘‘DDoS attack trends for 2022 Q1,’’ Cloudflare, CA, USA, Tech. Rep., Apr. 2022.
4. T. Shorey, D. Subbaiah, A. Goyal, A. Sakxena, and A. K. Mishra, ‘‘Performance comparison and analysis of Slowloris, GoldenEye and Xerxes DDoS attack tools,’’ in Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI), Sep. 2018, pp. 318–322.
5. L. F. Eliyan and R. Di Pietro, ‘‘DoS and DDoS attacks in software defined networks: A survey of existing solutions and research challenges,’’ Future Gener. Comput. Syst., vol. 122, pp. 149–171, Sep. 2021.
6. J. Mirkovic and P. Reiher, ‘‘A taxonomy of DDoS attack and DDoS defense mechanisms,’’ ACM SIGCOMM Comput. Commun. Rev., vol. 34, no. 2, pp. 39–53, Apr. 2004.
7. G. Somani, M. S. Gaur, D. Sanghi, M. Conti, and R. Buyya, ‘‘DDoS attacks in cloud computing: Issues, taxonomy, and future directions,’’ Comput. Commun., vol. 107, pp. 30–48, Jul. 2017.
8. A. Bhardwaj, G. V. B. Subrahmanyam, V. Avasthi, H. Sastry, and S. Goundar, ‘‘DDoS attacks, new DDoS taxonomy and mitigation solutions—A survey,’’ in Proc. Int. Conf. Signal Process., Commun., Power Embedded Syst. (SCOPES), Oct. 2016, pp. 793–798.
9. M. Sauter, ‘‘‘LOIC will tear us apart’ the impact of tool design and media portrayals in the success of activist DDOS attacks,’’ Amer. Behav. Scientist, vol. 57, no. 7, pp. 983–1007, 2013.
10. M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis, and D. Kumar, ‘‘Understanding the Mirai Botnet,’’ in Proc. 26th USENIX Secur. Symp., 2017, pp. 1093–1110.
11. B. Nagpal, P. Sharma, N. Chauhan, and A. Panesar, ‘‘DDoS tools: Classification, analysis and comparison,’’ in Proc. 2nd Int. Conf. Comput. Sustain. Global Develop. (INDIACom), Mar. 2015, pp. 342–346.
12. P. J. Shinde and M. Chatterjee, ‘‘A novel approach for classification and detection of DOS attacks,’’ in Proc. Int. Conf. Smart City Emerg. Technol. (ICSCET), Jan. 2018, pp. 1–6.
13. H. Beitollahi, D. M. Sharif, and M. Fazeli, ‘‘Application layer DDoS attack detection using cuckoo search algorithm-trained radial basis function,’’ IEEE Access, vol. 10, pp. 63844–63854, 2022.
14. D. Kshirsagar and J. M. Shaikh, ‘‘Intrusion detection using rule-based machine learning algorithms,’’ in Proc. 5th Int. Conf. Comput., Commun., Control Autom. (ICCUBEA), Sep. 2019, pp. 1–4.
15. W. Zhijun, L. Wenjing, L. Liang, and Y. Meng, ‘‘Low-rate DoS attacks, detection, defense, and challenges: A survey,’’ IEEE Access, vol. 8, pp. 43920–43943, 2020.
16. Z. Wu, Q. Pan, M. Yue, and L. Liu, ‘‘Sequence alignment detection of TCPtargeted synchronous low-rate DoS attacks,’’ Comput. Netw., vol. 152, pp. 64–77, Apr. 2019.
17. O. Boyar, M. E. Özen, and B. Metin, ‘‘Detection of denial-of-service attacks with SNMP/RMON,’’ in Proc. IEEE 22nd Int. Conf. Intell. Eng. Syst. (INES), Jun. 2018, pp. 000437–000440.
18. R. SaiSindhuTheja and G. K. Shyam, ‘‘An efficient metaheuristic algorithm based feature selection and recurrent neural network for DoS attack detection in cloud computing environment,’’ Appl. Soft Comput., vol. 100, Mar. 2021, Art. no. 106997.
19. S. Ramesh, C. Yaashuwanth, K. Prathibanandhi, A. R. Basha, and T. Jayasankar, ‘‘An optimized deep neural network based DoS attack detection in wireless video sensor network,’’ J. Ambient Intell. Hum. Comput., pp. 1–14, 2021.
20. T. A. Tuan, H. V. Long, L. H. Son, R. Kumar, I. Priyadarshini, and N. T. K. Son, ‘‘Performance evaluation of Botnet DDoS attack detection using machine learning,’’ Evol. Intell., vol. 13, no. 2, pp. 283–294, Jun. 2020.
21. P. Kumari and A. K. Jain, ‘‘A comprehensive study of DDoS attacks over IoT network and their countermeasures,’’ Comput. Secur., vol. 127, Apr. 2023, Art. no. 103096.
22. S. Wankhede and D. Kshirsagar, ‘‘DoS attack detection using machine learning and neural network,’’ in Proc. 4th Int. Conf. Comput. Commun. Control Autom. (ICCUBEA), Aug. 2018, pp. 1–5.
23. Y. Jia, F. Zhong, A. Alrawais, B. Gong, and X. Cheng, ‘‘FlowGuard: An intelligent edge defense mechanism against IoT DDoS attacks,’’ IEEE Internet Things J., vol. 7, no. 10, pp. 9552–9562, Oct. 2020.
24. I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, ‘‘Toward generating a new intrusion detection dataset and intrusion traffic characterization,’’ in Proc. 4th Int. Conf. Inf. Syst. Secur. Privacy, vol. 1, Jan. 2018, pp. 108–116.
25. F. Ridzuan and W. M. N. Wan Zainon, ‘‘A review on data cleansing methods for big data,’’ Proc. Comput. Sci., vol. 161, pp. 731–738, Jan. 2019.
26. J. Cai, J. Luo, S. Wang, and S. Yang, ‘‘Feature selection in machine learning: A new perspective,’’ Neurocomputing, vol. 300, pp. 70–79, Jul. 2018.
27. M. Ahsan, M. Mahmud, P. Saha, K. Gupta, and Z. Siddique, ‘‘Effect of data scaling methods on machine learning algorithms and model performance,’’ Technologies, vol. 9, no. 3, p. 52, Jul. 2021.
28. A. M. Mahfouz, D. Venugopal, and S. G. Shiva, ‘‘Comparative analysis of ML classifiers for network intrusion detection,’’ in Proc. 4th Int. Congr. Inf. Commun. Technol. Cham, Switzerland: Springer, 2020, pp. 193–207.
29. M. Al-Zewairi, S. Almajali, and A. Awajan, ‘‘Experimental evaluation of a multi-layer feed-forward artificial neural network classifier for network intrusion detection system,’’ in Proc. Int. Conf. New Trends Comput. Sci. (ICTCS), Oct. 2017, pp. 167–172.
30. D. Hunter, H. Yu, M. S. Pukish, III, J. Kolbusz, and B. M. Wilamowski, ‘‘Selection of proper neural network sizes and architectures—A comparative study,’’ IEEE Trans. Ind. Informat., vol. 8, no. 2, pp. 228–240, May 2012.

31. M. J. Madić and M. R. Radovanović, ‘‘Optimal selection of ANN training and architectural parameters using Taguchi method: A case study,’’ FME Trans., vol. 39, no. 2, pp. 79–86, 2011.
32. G. Panchal, A. Ganatra, Y. P. Kosta, and D. Panchal, ‘‘Behaviour analysis of multilayer perceptronswith multiple hidden neurons and hidden layers,’’ Int. J. Comput. Theory Eng., pp. 332–337, 2011.
33. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, ‘‘Scikit-learn: Machine learning in Python,’’ J. Mach. Learn. Res., vol. 12, no. 10, pp. 2825–2830, Jul. 2017.
34. L. Bottou, ‘‘Stochastic gradient learning in neural networks,’’ Proc. NeuroNimes, vol. 91, no. 8, p. 12, Nov. 1991.
35. R. Wu, H. Huang, X. Qian, and T. Huang, ‘‘A L-BFGS based learning algorithm for complex-valued feedforward neural networks,’’ Neural Process. Lett., vol. 47, no. 3, pp. 1271–1284, Jun. 2018.
36. A. S. Berahas and M. Takác, ‘‘A robust multi-batch L-BFGS method for machine learning,’’ Optim. Methods Softw., vol. 35, no. 1, pp. 191–219, Jan. 2020.
37. I. K. M. Jais, A. R. Ismail, and S. Q. Nisa, ‘‘Adam optimization algorithm for wide and deep neural network,’’ Knowl. Eng. Data Sci., vol. 2, no. 1, pp. 41–46, 2019.
38. D. P. Kingma and J. Ba, ‘‘Adam: A method for stochastic optimization,’’ 2014, arXiv:1412.6980.