ZKG INTERNATIONAL ISSN: 2366-1313

DETECTION OF APPLICATION LAYER DDOS ATTACKS PRODUCED BY VARIOUS FREELY ACCESSIBLE TOOLKITS USING MACHINE LEARNING

[1]M. Jhansirani,[2]M.Sanju Sandesh,[3]SK.Mehataj,[4]M.Dhinakar,[5]N.Vamshi,

1Assistant Professor, Dept. of IT, TKR College of Engineering and Technology, Meerpet, Hyderabad,

Jhansirani512@gmail.com

[2,3,4,5]BTech Student, Dept. of IT, TKR College of Engineering and Technology, Meerpet, Hyderabad

sanjumunugapati@gmail.com, mehatajshaik870@gmail.com, mandhadhinakar@gmail.com, vamshinadimpalli91@gmail.com

Abstract: The project aims to detect and mitigate escalating application-layer DDoS attacks, providing insights into attack patterns and tools for enhanced cybersecurity measures. With a target on HTTP-layer attacks, the project seeks to unravel tactics and tools, offering a specialized approach to bolster understanding and countermeasures against evolving cyber threats. There is an urgent need to address rising DDoS threats by shifting the project focus to tools' accessibility. This is crucial for proactive defense against the widespread use of malicious attack tools. The project aims to empower network administrators and cybersecurity experts, securing online services. Ultimately, it benefits users and businesses with resilient defenses against evolving DDoS threats. To boost performance, we introduced ensemble models—Voting Classifier (RandomForest, DecisionTree) and Stacking Classifier (RandomForest, DecisionTree, LGBM). These enhancements aim to improve cyberbullying detection accuracy.

Index terms - *DDoS, DDoS tools, machine learning, deep learning.*

1. INTRODUCTION

One of the most pernicious and increasingly complex security dangers to computer networks is distributed denial of service (DDoS) attacks [1], [2]. In Q1 2022, there was a significant increase in application-layer attacks, specifically HTTP-layer DDoS attacks which rose by 164% YoY and 135% QoQ. In terms of industry attacks, the Consumer Electronics sector experienced the highest increase with a staggering 5,086% QoQ. Online Media ranked second with a 2,131% increase in attacks QoQ, while Computer Software companies came in third with a 76% QoQ and 1,472 YoY increase in attacks [2].

A DDoS attack is a malevolent effort to stop a specific website, computer, or network from operating normally by saturating it with traffic from numerous sources. In this kind of attack, the perpetrator employs a network of computers or other

ISSN: 2366-1313

devices (referred to as a ‘‘botnet’’) to overwhelm the target system with an excessive quantity of data, rendering it inaccessible to authorized users [4]. DDoS attacks are complex attacks since (1) they can generate a large volume of traffic from a wide variety of sources, and (2) the traffic appears to originate from a wide variety of locations [5].

DDoS attacks manifest in diverse forms, with application layer attacks being one of them. Application-layer DDoS attacks target the application layer of the victim system, aiming to exhaust its resources or cause the application to fail. Illustrative examples of such attacks include HTTP floods and Slowloris attacks. The primary goal of application-layer DDoS attacks is to disable a network by overwhelming it with traffic, leading to system crashes or unavailability [6], [7], [8].

A significant factor in the growth of DDoS attacks is the easy availability of DDoS attack tools. These tools can be used intentionally to overwhelm servers and websites with traffic to the point where they become inoperable. Due to the easy availability of tools, either through purchasing them on the dark web or downloading freely accessible scripts, people with little to no technical knowledge can carry out devastating DDoS attacks [4], [9], [10], [11].

2. LITERATURE SURVEY

From smart home to industrial automation to smart power grid, IoT-based solutions penetrate into every working field. These devices expand the attack surface and turned out to be an easy target for the attacker as resource constraint nature hinders the integration of heavy security solutions. Because IoT devices are less secured and operate mostly in unattended scenario, they perfectly justify the requirements of attacker to form botnet army to trigger Denial of Service attack on massive scale [1, 2, 17]. Therefore, this paper [1] presents a Machine Learning-based attack detection approach to identify the attack traffic in Consumer IoT (CIoT). This approach operates on local IoT network-specific attributes to empower low-cost machine learning classifiers to detect attack, at the local router. The experimental outcomes unveiled that the proposed approach achieved the highest accuracy of 0.99 which confirms that it is robust and reliable in IoT networks.

Today Internet is becoming an emerging technology for remote control of industrial applications, where one site needs to control another site remotely (e.g. power plants controllers). Denial-of-Service (DoS) attacks may cause significant disruptions to the Internet which will threaten the operation of such network based control systems. Overlay networks have been proposed to protect Internet application sites by location-hiding technique. This paper [2] analyzes a large domain of previous approaches against this problem. This paper addresses how an interface to an overlay network can be designed such that communication services among geographically distributed application sites are secured against DoS attacks. This paper presents a novel architecture called overlay protection layer (OPL) that proactively protect application sites from DoS attacks [17, 18]. Through simulation this paper shows DoS attacks have a negligible chance to disrupt communications services via the OPL architecture. Even if attackers attack 50% of overlay nodes via a Distributed DoS attack still 75% of communication channels are available.

ISSN: 2366-1313

DDoS attack has been the most preferred attack by the hackers in the recent years. This is due to its ability to create multitude and variety of problems. A large group of hackers and experts in this field have developed packages and tools that initiate DDoS attack on various type of networks. It is essential to evaluate and compare the strength of DDoS attack launched by these tools to devise efficient countermeasures against it. In this paper [4], the performance of three DDoS attack tools is compared and analyzed using parameters such as time to successfully launch attack, traffic rate, and packet size. [20] The DDoS tools considered for evaluation are Slowloris, GoldenEye and Xerxes. The experimental results infer that Xerxes outperforms other tools in launching a DDoS attack[21, 23].

Software Defined Networking (SDN) is a new networking paradigm where forwarding hardware is decoupled from control decisions. It promises to dramatically simplify network management and enable innovation and evolution. In SDN, network intelligence is logically centralized in software-based controllers (the control plane), while network devices (OpenFlow Switches) become simple packet-forwarding devices (the data plane) that can be programmed via an open interface (OpenFlow protocol). Such decoupling of the control plane from the data plane introduces various challenges that include security, reliability, load balancing, and traffic engineering. Dreadful security challenges in SDNs are denial of service (DoS) and distributed denial of service (DDoS) attacks [5]. For instance, in SDNs, DoS/DDoS attacks could flood the control plane, the data plane, or the communication channel. Attacking the control plane could result in failure of the entire network, while attacking the data plane or the communication channel results in packet drop and network unavailability. In this paper we deliver several contributions that shed light on the field of DoS/DDoS attacks in SDNs, providing a complete background about the area, including attacks and analysis of the existing solutions [19, 20]. In particular, our contributions can be summarized as follow: we review and systematize the state-of-the-art solutions that address both DoS and DDoS attacks in SDNs through the lenses of intrinsic and extrinsic approaches. Moreover, the discussed countermeasures are organized accordingly to their focus, be it on detection, mitigation, prevention, or graceful degradation. Further, we survey the different approaches and tools adopted to implement the revised solutions. Finally, we also highlight possible future research directions to address DoS/DDoS attacks in SDNs [21].

Distributed denial-of-service (DDoS) is a rapidly growing problem. The multitude and variety of both the attacks and the defense approaches is overwhelming. This paper [6] presents two taxonomies for classifying attacks and defenses, and thus provides researchers with a better understanding of the problem and the current solution space. The attack classification criteria was selected to highlight commonalities and important features of attack strategies, that define challenges and dictate the design of countermeasures. The defense taxonomy classifies the body of existing DDoS defenses based on their design decisions; it then shows how these decisions dictate the advantages and deficiencies of proposed solutions [4, 23].

3. METHODOLOGY

i) Proposed Work:

ISSN: 2366-1313

The proposed system for DDoS attack detection offers a more comprehensive approach, moving beyond just recognizing attack patterns [20]. Instead, we consider the bigger picture by assessing the accessibility and impact of widely available attack tools. This broader perspective helps improve our understanding of cybersecurity threats and enables the development of more effective defenses. The system is adaptable to changing threats, avoiding a narrow focus on specific attack types. In an effort to enhance performance, we integrated advanced ensemble models into the project, including a Voting Classifier combining RandomForest and DecisionTree, as well as a Stacking Classifier with RandomForest and DecisionTree as base learners [20]. This ensemble approach aims to improve cyberbullying detection accuracy. Additionally, a user-friendly Flask framework with SQLite was implemented for secure signup and signin, facilitating user testing by providing input and obtaining results. These extensions not only diversify model architectures for heightened accuracy but also streamline user interactions, contributing to the project's robustness and practical usability.

ii) System Architecture:

The system architecture begins with data preparation, utilizing the NSL-KDD [13] and NBOT-IOT datasets. Feature selection follows, optimizing the data for efficient analysis. Subsequently, three classifiers are employed: MLP with diverse optimization techniques (SGD, LBFGS, Adam), Extension Stacking Classifier, and Extension Voting Classifier. These classifiers collectively enhance prediction accuracy. The comprehensive architecture ensures robust analysis and prediction, making it a versatile and effective system for identifying and countering DDoS attacks in network security.

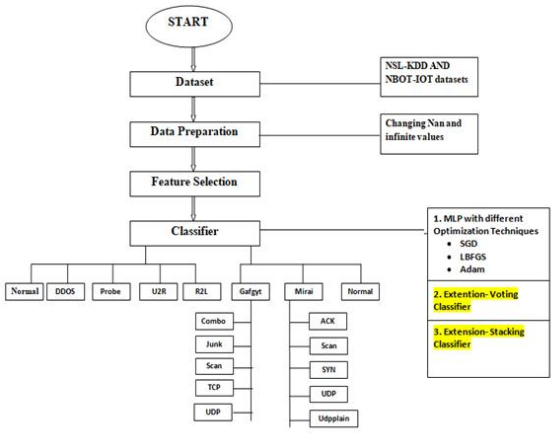


Fig 1 Proposed architecture

iii) Dataset collection:

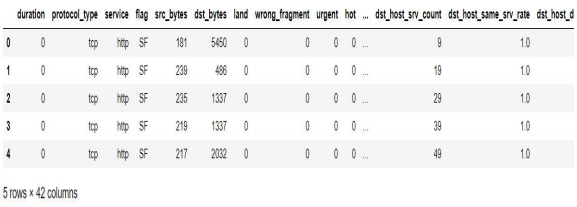
NSL-KDD Dataset:

The NSL-KDD dataset is a benchmark dataset widely used for evaluating intrusion detection systems. It is an improved version of the original KDD Cup 99 dataset, addressing its limitations. NSL-KDD offers a diverse set of network traffic data, including normal and various types of attacks, making it suitable for training and testing machine learning models in the field of cybersecurity [13].

NBOT-IOT Dataset:

The NBOT-IOT dataset focuses on network behavior analysis for the Internet of Things (IoT). It comprises data generated by various IoT devices, providing insights into the communication patterns and potential threats within IoT networks. This dataset is essential for developing machine learning models tailored to detect anomalies and potential cyber threats specific to IoT environments.

ISSN: 2366-1313



	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 involves transforming raw data into valuable information for businesses. Generally, data scientists process data, which includes collecting, organizing, cleaning, verifying, analyzing, and converting it into readable formats such as graphs or documents. Data processing can be done using three methods i.e., manual, mechanical, and electronic. The aim is to increase the value of information and facilitate decision-making. This enables businesses to improve their operations and make timely strategic decisions. Automated data processing solutions, such as computer software programming, play a significant role in this. It can help turn large amounts of data, including big data, into meaningful insights for quality management and decision-making.

v) Feature selection:

Feature selection is the process of isolating the most consistent, non-redundant, and relevant features to use in model construction. Methodically reducing the size of datasets is important as the size and variety of datasets continue to grow. The main goal of feature selection is to improve the performance of a predictive model and reduce the computational cost of modeling.

Feature selection, one of the main components of feature engineering, is the process of selecting the most important features to input in machine learning algorithms. Feature selection techniques are employed to reduce the number of input variables by eliminating redundant or irrelevant features and narrowing down the set of features to those most relevant to the machine learning model. The main benefits of performing feature selection in advance, rather than letting the machine learning model figure out which features are most important.

vi) Algorithms:

In the project, **the Multilayer Perceptron (MLP)** with the Stochastic Gradient Descent (SGD) optimization algorithm is utilized for its effectiveness in training neural networks. SGD is a variant of gradient descent that randomly selects a subset of training samples, or a batch, for each iteration, making it computationally efficient. The combination of MLP and SGD is well-suited for this project as it efficiently learns and adapts to complex patterns in the NSL-KDD and NBOT-IOT datasets, providing a robust foundation for DDoS attack detection through its ability to navigate high-dimensional feature spaces [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): Limited-memory Broyden-Fletcher-Goldfarb-Shanno

ISSN: 2366-1313

(lbfgs) is a quasi-Newton optimization algorithm designed for unconstrained optimization problems. lbfgs belongs to the family of quasi-Newton methods, which aim to find the minimum of a function without explicitly computing its derivatives. lbfgs maintains an approximation of the inverse Hessian matrix to iteratively update the model parameters. It is particularly effective in scenarios where the dataset is not extremely large, and the model has a moderate number of parameters. For THIS project, if the dataset is of moderate size and the model has a reasonable number of parameters, lbfgs could be a suitable choice. It tends to converge faster than some other optimization algorithms, especially in scenarios where the data fits well into 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: Adam (Adaptive Moment Estimation) is an adaptive learning rate optimization algorithm that combines elements from both momentum and RMSprop. It adapts the learning rates of each parameter individually based on their past gradients, making it well-suited for scenarios with sparse gradients or noisy data. Adam is known for its efficiency and is often the default choice for many deep learning tasks. It includes mechanisms to control both the step size and the exponential decay of past gradients [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** combines predictions from base classifiers like Random Forest and Decision Tree, using a final estimator (LGBMClassifier). By leveraging diverse classifiers, it enhances accuracy and robustness against evolving DDoS attacks, making it well-suited for the project's network security goals.

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 predictions from a GridSearchCV-optimized Random Forest Classifier and a Decision Tree Classifier using a "soft" voting mechanism. This ensemble approach enhances predictive performance by considering weighted averages of class probabilities, making it valuable for DDoS attack detection. Leveraging diverse classifiers strengthens defense against a range of cyber threats, ensuring a more resilient and effective security strategy.

ISSN: 2366-1313

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: Precision evaluates the fraction of correctly classified instances or samples among the ones classified as positives. Thus, the formula to calculate the precision is given by:

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

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

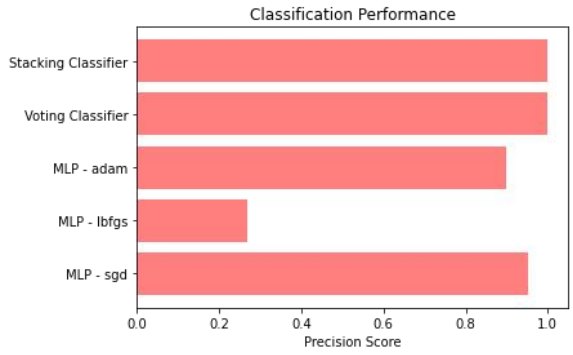


Fig 6 Precision comparison graph

Recall: Recall is a metric in machine learning that measures the ability of a model to identify all relevant instances of a particular class. It is the ratio of correctly predicted positive observations to the total actual positives, providing insights into a model's completeness in capturing instances of a given class.

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

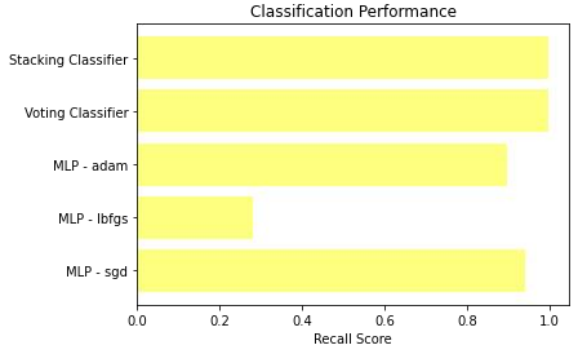


Fig 7 Recall comparison graph

Accuracy: Accuracy is the proportion of correct predictions in a classification task, measuring the overall correctness of a model's predictions.

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

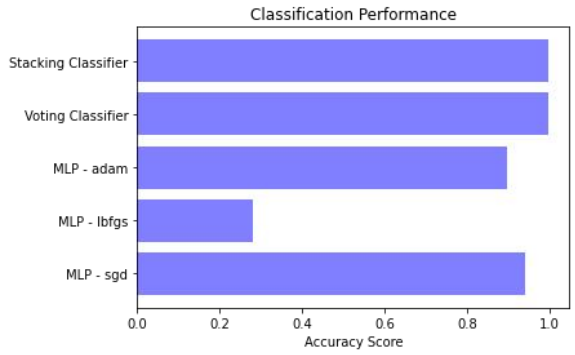


Fig 8 Accuracy graph

ISSN: 2366-1313

F1 Score: The F1 Score is the harmonic mean of precision and recall, offering a balanced measure that considers both false positives and false negatives, making it suitable for imbalanced datasets.

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

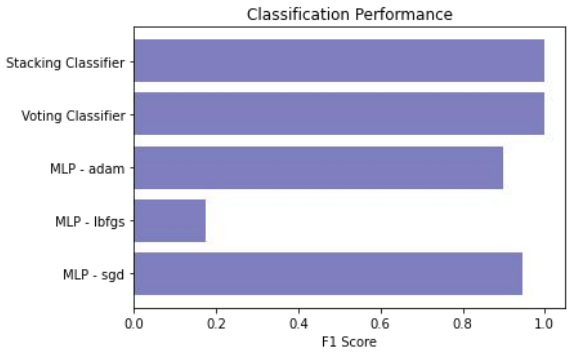


Fig 9 F1Score

	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

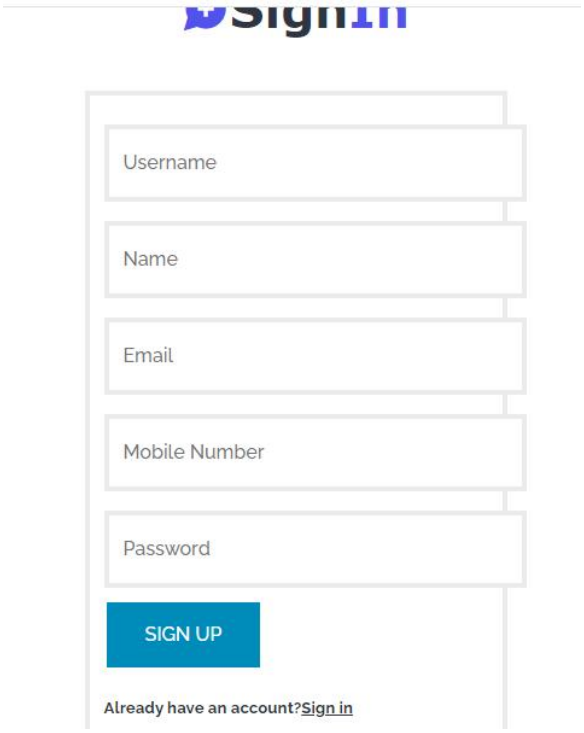


Fig 12 Signin page

Fig 13 Login page

ISSN: 2366-1313

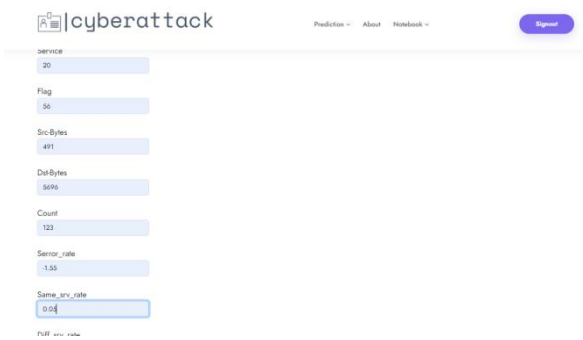


Fig 14 User input

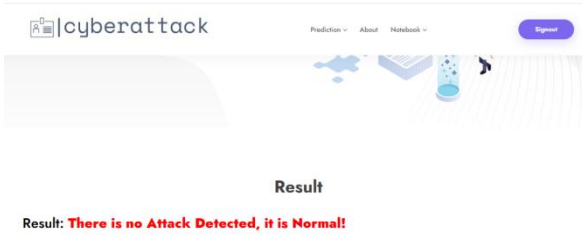


Fig 15 Predict result for given input

5. CONCLUSION

The project significantly contributes to cybersecurity by developing advanced techniques for detecting and mitigating Distributed Denial of Service (DDoS) attacks, bolstering the resilience of computer networks [16, 17]. Through thorough exploration of diverse datasets (KDD-CUP and NBOT-IOT), valuable insights into network traffic characteristics and potential attack patterns were gained, providing a foundation for effective model development. The evaluation of Multi-Layer Perceptron (MLP) [21] models with different optimizers, including SGD, lbfgs, and adam, identified the most effective approach for DDoS attack detection, contributing to the robustness of cybersecurity measures. And also added voting and stacking classifiers, combining predictions from multiple models, showcase innovation. This approach enhances prediction accuracy and resilience against diverse cyber threats. The integration of a Flask framework with SQLite for user signup and signin, coupled with a user-friendly front-end, ensures practical applicability. Users can conveniently provide input, witness predictions, and interact with the system, enhancing real-world usability.

6. FUTURE SCOPE

Future advancements in machine learning can be integrated into the project to enhance the accuracy and efficiency of DDoS attack detection [21]. Exploring and implementing state-of-the-art algorithms and models may further strengthen the system's ability to adapt to evolving cyber threats. The project's future scope includes the development of real-time detection mechanisms and responsive strategies. Integrating technologies that enable swift identification and mitigation of DDoS attacks as they occur will be crucial for minimizing the impact of such threats on network resources [23]. Incorporating advanced behavioral analysis techniques can contribute to the project's future development. By studying the normal behavior of networks and devices, the system can more effectively identify anomalies associated with DDoS attacks, enabling a proactive approach to cybersecurity. As networks and cyber threats continue to evolve, the project's future scope involves ensuring scalability and adaptability. Designing the system to handle larger datasets, diverse attack patterns, and emerging technologies will be essential to maintain its effectiveness in the ever-changing landscape of cybersecurity.

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.