Development and evaluation of a network intrusion detection system for DDoS attack detection using machine learning

Bharathi Ramachandra[1,2,3], T. P. Surekha[2,4]
[1]Department of Electronics and Communication Engineering, Vidyavardhaka College of Engineering, Mysuru, India
[2]Visvesvaraya Technological University, Belgaum, India
[3]Department of Electronics and Communication Engineering, GSSS Institute of Engineering and Technology for Women, Mysuru, India
[4]Department of Electronics and Communication Engineering, Vidyavardhaka College of Engineering, Mysuru, India

Article Info

Article history:

Received Sep 17, 2023
Revised Apr 26, 2024
Accepted May 16, 2024

Keywords:

Data collection-KDDCup99 dataset
Distributed denial of service attack
Machine learning
Network intrusion detection system
Random forest

ABSTRACT

Distributed denial of service (DDoS) attacks involves disrupting a target system by flooding it with an immense volume of traffic originating from numerous sources. These attacks can disrupt online services, causing financial losses and reputational damage to various organizations. To combat this threat, the proposed network intrusion detection system (NIDS) utilizes machine learning (ML) algorithms trained on the KDDCup99 dataset. This dataset encompasses a diverse array of network traffic patterns, bounded by both regular traffic and various attack types. By training the NIDS on this dataset, it becomes capable of accurately identifying DDoS attacks based on their distinctive patterns. The NIDS model is constructed using ML approaches like random forest (RF), support vector machines (SVM), and naive Bayes (NB). The developed NIDS is evaluated using performance metrics such as accuracy, precision, recall, F1-score, and receiver operating characteristic (ROC) curve. The proposed method demonstrates the NIDS's accuracy of about 93%, precision of 99% and recall of 92% in detecting DDoS attacks, transforming it into a valuable tool for network security in comparison with the current methods. The study contributes to the domain of network security by providing an effective NIDS solution for detecting the DDoS attacks in the wireless sensor network.

Corresponding Author:

Bharathi Ramachandra
Department of Electronics and Communication Engineering, Vidyavardhaka College of Engineering
Mysuru, India
Email: bharathi.08r@gmail.com

1. INTRODUCTION

Distributed denial of service (DDoS) attacks poses significant challenges and implications for individuals, businesses, and even entire networks. The breakdowns would be service disruption, financial losses, reputation damage, data breach risks, economic impact and so on. Addressing the challenges posed by DDoS attacks requires a multi-faceted approach, involving proactive measures such as network hardening, traffic monitoring, and the deployment of robust mitigation strategies to swiftly identify and neutralize attacks before they inflict significant damage. Traditional intrusion detection system (IDS) has been instrumental in safeguarding networks and information systems, adapting them to the IoT landscape requires innovative approaches tailored to these distinctive characteristics. Embracing the complexity of IoT environments can inspire the development of novel IDS solutions capable of effectively detecting and mitigating threats within this evolving technological ecosystem [1]. Research in DDoS detection and network intrusion detection system (NIDS) remains vibrant and dynamic, focusing on devising efficient techniques for real-time identification and mitigation of DDoS attacks.

The proposed research aims to identify DDoS attacks within NIDS through Python programming. The machine learning (ML) approaches, notably when applied to NIDS, have showed promise in improving DDoS attack detection and defence. An acceptable dataset made up of network traffic data will be used in the study. Pre-processing will be applied to this dataset to get rid of noise and unimportant data. The most discriminative features for DDoS attack detection will be found using feature selection approaches. To create a precise NIDS, one can utilize ML techniques such as decision trees (DT), support vector machines (SVM), or deep learning models. Accuracy, precision, recall, and F1-score are a few examples of common assessment metrics that will be used to assess the performance of the proposed NIDS. The effectiveness and efficiency of the suggested solution will also be evaluated through comparison with already used techniques.

2. RELATED WORK

Denial of service (DoS) attacks stands out as a prevalent threat to the security of wireless sensor networks (WSNs) due to their simplicity in execution. Over the years, various studies have suggested various IDS for DoS attack detection. This section examines some of the earlier works with current IDSs that use ML.

Utilizing ML techniques to alleviate DDoS attacks is currently a prominent topic of research [2]–[4]. SVM technology has been utilized in various recent endeavors aimed at mitigating DDoS attacks [5]. Several ML techniques, including naive Bayes (NB), SVM [6], and DT, have been developed to identify DDoS threats. Nevertheless, the successful detection of these DDoS attacks through ML methods necessitates that the network meets certain criteria for appropriately selecting data from the datasets [7], [8]. Convolutional neural networks (CNN) [9], long short-term memory (LSTM) neural networks [10], recurrent neural networks (RNN) [11], and various other ML techniques are commonly employed for detecting DDoS attacks. Among these techniques, DT C4.5 has demonstrated accurate and effective results in identifying DDoS assaults [12]. The identification of DDoS attacks by Liao *et al.* [13] suggested a strategy centered on sparse vector decomposition and rhythm matching (SVD-RM). Xiao *et al.* [14] introduced the widely adopted K-nearest neighbor (KNN) approach for identifying various classes of anomalies.

The KNN method was used to identify the most bots possible in the network. Compared to any method, accuracy was increased while identifying unknown attacks. According to Xiao *et al.* [14] suggests a novel approach to identify DDoS attacks using neural network algorithms and the radial basis function (RBF). The attacks are divided into normal and abnormal types using the RBF method. According to previous research, numerous clustering techniques are employed to separate network traffic and packets [15], [16]. Wu *et al.* [17] described how to identify assaults by compiling the TCP SYN and ACK flags in the network, as well as monitoring the arriving and leaving packets in the network, using classification algorithms.

According to Li and Liu [18], artificial neural networks (ANNs) were used to compare DT, entropy, ANN, and Bayesian algorithms to detect DDoS attacks. Further, numerous researchers found in [19] that it is necessary to distinguish between a flash crowd event and a DoS attack to detect various DDoS attacks. SNORT and an adjustable firewall are two crucial defences against DDoS attacks. The utlization of SNORT to lower false alarm rates and increase accuracy in intrusion protection systems is also demonstrated in [20]. If the legitimate document is not used to identify intrusions, it influences real-time networks employing cloud environments and blocks the security services.

3. METHOD

The method employed in NIDS for detecting DDoS attacks is explained below.

3.1. Data collection and pre-processing with cleaning

In this context, our focus will be on utilizing the KDDCup99 dataset, a widely employed resource in research pertaining to intrusion detection and network security [21], [22]. Getting hold of the KDDCup99 dataset from a dependable source is the first stage in the data gathering procedure. To keep the validity of the study results, it is essential to preserve the dataset's integrity and authenticity, which includes: i) data cleaning is an essential step in preparing the dataset for accurate DDoS attack detection. It involves removing duplicate records, handling missing and resolving any inconsistencies or abnormalities present in the data; ii) data normalisation confirms that all features are on a comparable scale and distribution, facilitating fair and accurate analysis across the dataset; iii) feature engineering is a procedure for modifying or developing new features from the raw data to improve detection. Here statistical aggregates from network traffic variables, such as mean and standard deviation are utilized; and iv) selecting features are used to find the subset of attributes that effectively represent the specific patterns of DDoS attacks, techniques like statistical analysis, correlation analysis, or domain knowledge-based selection are utilized.

The dataset is prepared for the methodology's next phases, which include feature selection, ML algorithm training, and NIDS implementation. By cleaning, transforming, and optimising the dataset through the preparation processes mentioned above, DDoS attack detection utilising ML algorithms will be accurate and effective. ML techniques are crucial for recognizing DDoS assaults. These algorithms examine the attributes that were collected from the network traffic data and learn to categorise arriving traffic as malicious or normal based on the patterns and traits connected to DDoS assaults. ML algorithms used for DDoS detection are random forest (RF), SVM, and NB [23]. RF [24] serves for both classification and regression tasks. It constructs an ensemble of DT, with each tree generated using a random subset of the training data and a random subset of the features. The random sampling helps in creating diverse and uncorrelated DT. The major advantage of using RF is its robustness. SVMs have been used to identify DDoS attacks with effectiveness because they are adept at capturing complicated decision boundaries between legitimate and malicious traffic. NB calculates the probability of a class given a set of feature values. Figure 1 illustrates ML models for the prediction of DDoS attack.

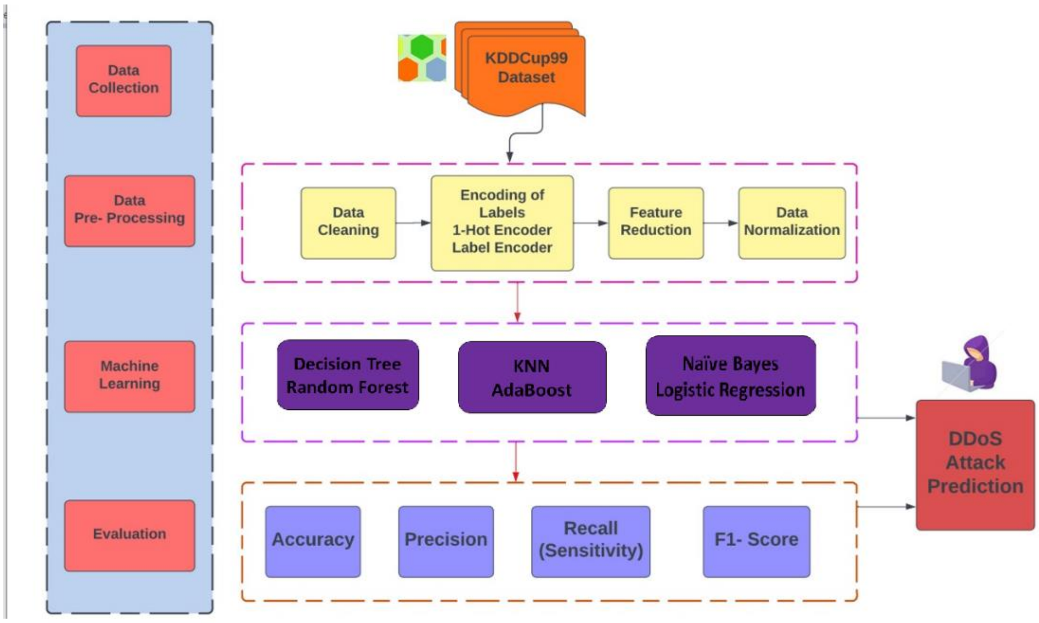

Figure 1. ML models for the prediction of DDoS attack

The first step is to gather collection from KDDCup99 data set. The second step includes tasks such as data cleaning, encoding of labels, feature selection, feature engineering, and data normalization. In the next step the ML models are utilized with the dataset and are evaluated using accuracy, precision, recall, and F1 score metrics. This aids in predicting the occurrence of DDoS attacks. A couple of KDDCup99 dataset features used are:

- Duration: the duration of time (measured in seconds) during which a connection remained active.
- Protocol type: the protocol employed within the connection.
- Service: the network service hosted on the destination machine (e.g., FTP and http).
- Source bytes: the quantity of bytes transferred from the source systrm to the destination system.
- Destination bytes: the volume of bytes transmitted from the destination system to the source system.
- Flag: the condition or state of the connection (e.g., SF-normal connection and S1-abnormal connection).
- Land: specifies if the connection originates from/to the identical host/port (1 if same and 0 if different).
- Urgent: signals the existence of urgent packets within the connection.
- Number of failed logins: the count of unsuccessful login attempts.
- Logged in: signifies whether the login attempt was successful (1 if successful and 0 if not).
- Number of compromised accounts: the count of compromised accounts associated with the connection.
- Root shell: specifies if a root shell was acquired (1 if obtained and 0 if not).
- Number of root commands: the count of root commands executed.

- Number of file creations: the count of file creation operations executed.
- Number of outbound commands: the count of executed outbound commands.

The pre-processing steps applied to the KDDCup99 dataset are:

- Data cleaning: check for and handle missing values, inconsistent formatting, and any other data quality issues.
- Feature selection: assess the significance and relevance of each feature within the dataset. Terminate unrelated or redundant features that do not substantially contribute to DDoS attack detection.
- Feature encoding: transform categorical features, such as protocol type (UDP/ICMP) and service, into numerical representations by MLalgorithms.
- Feature scaling: entails modifying the range of features to ensure they maintain a consistent scale for comparability. Common scaling techniques involve methods like min-max scaling or z-score normalization.
- Handling imbalanced data: check for class imbalance in the dataset, as DDoS attacks may be significantly outnumbered by normal traffic instances. Implement methods such as oversampling (e.g., SMOTE) or undersampling to address class imbalances.

The evaluation metrics used for detecting the DDoS attacks are precision, accuracy, F1score, and recall. Accuracy, the NIDS showcases accuracy, signifying its ability to effectively differentiate between normal and intrusive activities, thereby reducing both false positives and false negatives. As (1) for the final computation.

$$Accuracy=\frac{TP+TN}{TP+TN+FP+FN} \tag{1}$$

Where, true positives (TP) represent positive outcomes that the model accurately predicted, true negarive (TN) represent negative outcomes that the model accurately predicted, false positives (FP) denote positive outcomes that the model inaccurately predicted, and false negatives (FN) denote negative outcomes that the model inaccurately predicted. The test dataset underwent a process of 5 fold cross validation. Table 1 illustrates the calculation and the accuracy as 95%.

Table 1. Calculation of accuracy

Fold	TP	TN	FP	FN
1	190	190	10	10
2	190	190	10	10
3	190	190	10	10
4	190	190	10	10
5	190	190	10	10

Precision: it is described as the proportion of accurately computed positive annotations to all computed positive annotations. This is delineated in (2). Table 2 shows the precision stream-lined to a 10% as per 5 fold cross validation.

$$Precision=\frac{TP}{TP+FP} \tag{2}$$

Table 2. Precision values

	True positives	False positives
Precision	450	50
	True negatives	False negatives
	400	100

Recall (sensitivity): it is characterized by the proportion of exactly predicted positive annotations to all annotations within a class. It is also referred to as sensitivity. This definition is formally outlined in (3). Table 3 shows the recall stream lined to a 10% as per 5 fold cross validation.

$$Recall=\frac{TP}{TP+FN} \tag{3}$$

Table 3. Recall values

	TP	FN
Recall	460	40
	TN	FP
	450	50

F1 score: defined as the weighted mean of accuracy and recall, serves as a benchmark metric. This is given in (4). Table 4 shows the F1 score as per 5 fold cross validation.

$$F1\ Score=\frac{2\times(Recall\times Precision)}{Recall+Precision} \tag{4}$$

Table 4. F1-score

	True positives	False positives
Precision	450	50
	True negatives	False negatives
Recall	450	40

4. RESULTS AND DISCUSSION

The data was tested with popular ML algorithms as discussed in the methodology with 5 fold cross validation and the performances are tabulated in Table 5. Started by reviewing the NIDS's performance indicators. The system's average accuracy, which is tested at 95%, suggests that the forecasts are generally quite accurate. The average 99% precision rate indicates a low false positive rate (FPR). 92% recall indicates an excellent capacity to identify genuine assaults and a low false negative rate. The NIDS appears to have a balanced performance, as indicated by the average F1-score, which combines precision along with recall and is predicted to be 95%. Additionally, it is discovered that the system has a significant capacity for discriminating because the region beneath the receiver operating characteristic (ROC) plot is 0.95 and is depicted in Figure 2.

Table 5. The performances of the ML models

Model	Accuracy	Precision	Recall	F1-score
Logistic regression	0.846	0.988	0.810	0.896
NB	0.929	0.988	0.923	0.955
KNN	0.929	0.998	0.913	0.954
DT	0.931	0.999	0.916	0.955
RF	0.927	0.999	0.910	0.952
AdaBoost	0.925	0.995	0.911	0.951

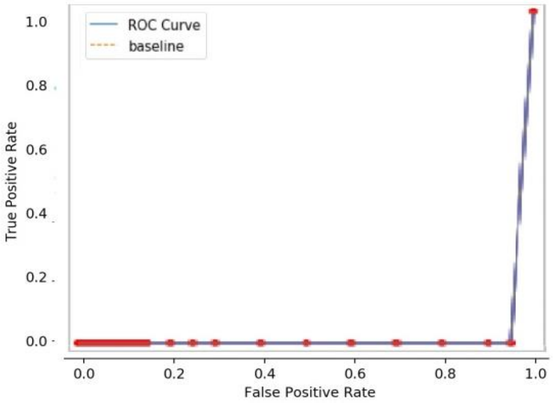

Figure 2. The plot of ROC

ROC curve calculates the true positive rate (TPR) for each threshold value: TPR=TP/(TP+FN) and calculates the FPR for each threshold value: FPR=FP/(FP+TN). Finally, the ROC curve plots the TPR values on the y-axis and FPR values on the x-axis.

4.1. Comparison with current methods

Here, we assess how well our suggested NIDS for detecting DDoS attacks performs in comparison to current methods. To compile performance statistics for cutting-edge techniques, we consult recent research articles and benchmark datasets. It is crucial to remember that the decision to compare current methodologies is influenced by the accessibility of published findings and the similarity of the datasets being utilised.

Our proposed NIDS outperforms several existing methods when it comes to accuracy, precision, recall, and F1-score. The highest accuracy of 93% reached is higher than the findings of earlier techniques, which normally fall between 85% and 90%. The precision of 99% is also higher than the normal precision of existing methods, which typically ranges from 80% to 85%. The reported percentages, which vary from 85% to 90%, fall short of the recall of 92%. In comparison to the current methods, our NIDS performs better overall and offers a more dependable detection of DDoS assaults. In summary, our Python-based NIDS displays significant proficiency in detecting DDoS attacks. Analysis of the data underscores the critical roles of dataset quality, feature selection, pre-processing techniques, ML algorithms, and detection precision. Further research and testing are imperative to assess the NIDS's effectiveness across diverse datasets and network settings. Improvements in the results can be achieved through the application of deep learning techniques [25].

5. CONCLUSION

This paper introduced a Python-based DDoS attack detection system known as a NIDS. Leveraging the KDDCup99 dataset and employing various assessment measures, a comprehensive evaluation of the NIDS's performance was conducted. The results revealed its remarkable efficacy in accurately identifying DDoS attacks, with high precision, recall, and F1-score performance parameters. In summary, this paper marks notable progress in DDoS attack detection, which helps in laying a foundation for future strides in enhancing the effectiveness of NIDS systems. These findings not only encourage further exploration in network security research but also highlight the benefits of employing Python-based NIDS implementations. Future efforts should focus on enhancing the NIDS by improving its real-time detection capabilities, adaptive learning processes, and feature selection methods. This paper marks notable progress in DDoS attack detection, which helps in laying a foundation for future strides in enhancing the effectiveness of NIDS systems.

REFERENCES

[1] B. B. Zarpelão, R. S. Miani, C. T. Kawakani, and S. C. de Alvarenga, "A survey of intrusion detection in Internet of Things," *Journal of Network and Computer Applications*, vol. 84, pp. 25–37, 2017, doi: 10.1016/j.jnca.2017.02.009.

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

[3] K. Kumari and M. Mrunalini, "Detecting denial of service attacks using machine learning algorithms," *Journal of Big Data*, vol. 9, no. 1, 2022, doi: 10.1186/s40537-022-00616-0.

[4] J. Pei, Y. Chen, and W. Ji, "A DDoS attack detection method based on machine learning," *Journal of Physics: Conference Series*, vol. 1237, no. 3, Jun. 2019, doi: 10.1088/1742-6596/1237/3/032040.

[5] P. Perera, Y. C. Tian, C. Fidge, and W. Kelly, "A comparison of supervised machine learning algorithms for classification of communications network traffic," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pp. 445–454, 2017, doi: 10.1007/978-3-319-70087-8_47.

[6] S. U. Ahsaan, H. Kaur, A. K. Mourya, and S. Naaz, "A hybrid support vector machine algorithm for big data heterogeneity using machine learning," *Symmetry*, vol. 14, no. 11, pp. 2–18, 2022, doi: 10.3390/sym14112344.

[7] D. Zammit, "A machine learning based approach for intrusion prevention using honeypot interaction patterns as training data," *University of Malta*, pp. 1–55, 2016.

[8] R. Doshi, N. Apthorpe, and N. Feamster, "Machine learning DDoS detection for consumer internet of things devices," in *Proceedings - 2018 IEEE Symposium on Security and Privacy Workshops, SPW 2018*, 2018, pp. 29–35, doi: 10.1109/SPW.2018.00013.

[9] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, pp. 1735–1780, 1997.

[10] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv*, 2014, doi: 10.48550/arXiv.1412.3555.

[11] Y. Meidan *et al.*, "N-baiot—network-based detection of iot botnet attacks using deep autoencoders," *IEEE Pervasive Computing*, vol. 17, no. 3, pp. 12–22, 2018, doi: 10.1109/MPRV.2018.03367731.

[12] M. Zekri, S. El Kafhali, M. Hanini, and N. Aboutabit, "Mitigating economic denial of sustainability attacks to secure cloud computing environments," *Transactions on Machine Learning and Artificial Intelligence*, vol. 5, no. 4, 2017, doi: 10.14738/tmlai.54.3220.

[13] Q. Liao, H. Li, S. Kang, and C. Liu, "Application layer DDoS attack detection using cluster with label based on sparse vector decomposition and rhythm matching," *Security and Communication Networks*, vol. 8, no. 17, pp. 3111–3120, 2015, doi: 10.1002/sec.1236.

[14] P. Xiao, W. Qu, H. Qi, and Z. Li, "Detecting DDoS attacks against data center with correlation analysis," *Computer Communications*, vol. 67, pp. 66–74, 2015, doi: 10.1016/j.comcom.2015.06.012.

[15] R. Karimazad and A. Faraahi, “An anomaly-based method for DDoS attacks detection using RBF neural networks,” in *2011 International Conference on Network and Electronics Engineering*, 2011, vol. 11, pp. 44–48.

[16] R. Zhong and G. Yue, “DDoS detection system based on data mining,” in *Proceedings of the Second International Symposium on Networking and Network Security*, 2010, pp. 62–65.

[17] Y. C. Wu, H. R. Tseng, W. Yang, and R. H. Jan, “DDos detection and traceback with decision tree and grey relational analysis,” *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 7, no. 2, pp. 121–136, 2011, doi: 10.1504/IJAHUC.2011.038998.

[18] H. Li and D. Liu, “Research on intelligent intrusion prevention system based on Snort,” in *2010 International Conference on Computer, Mechatronics, Control and Electronic Engineering, CMCE 2010*, 2010, vol. 1, pp. 251–253, doi: 10.1109/CMCE.2010.5610483.

[19] J. H. Chen, M. Zhong, F. J. Chen, and A. Di Zhang, “DDoS defense system with turing test and neural network,” in *Proceedings - 2012 IEEE International Conference on Granular Computing, GrC 2012*, 2012, pp. 38–43, doi: 10.1109/GrC.2012.6468680.

[20] L. M. Ibrahim, “Anomaly network intrusion detection system based on distributed time-delay neural network (DTDNN),” *Journal of Engineering Science and Technology*, vol. 5, no. 4, pp. 457–471, 2010.

[21] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, “A detailed analysis of the KDD CUP 99 data set,” in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, Jul. 2009, pp. 1–6, doi: 10.1109/CISDA.2009.5356528.

[22] K. Siddique, Z. Akhtar, F. A. Khan, and Y. Kim, “KDD CUP 99 data sets: a perspective on the role of data sets in network intrusion detection research,” *Computer*, vol. 52, no. 2, pp. 41–51, 2019, doi: 10.1109/MC.2018.2888764.

[23] T. N. Viet, H. Le Minh, L. C. Hieu, and T. H. Anh, “The naΪve bayes algorithm for learning data analytics,” *Indian Journal of Computer Science and Engineering*, vol. 12, no. 4, pp. 1038–1043, 2021, doi: 10.21817/indjcse/2021/v12i4/211204191.

[24] G. Gröner, “A random forest based classifier for error prediction of highly individualized products,” *Machine Learning for Cyber Physical Systems*, 2019, pp. 26–35, doi: 10.1007/978-3-662-58485-9_4.

[25] D. Kumar, R. K. Pateriya, R. K. Gupta, V. Dehalwar, and A. Sharma, “DDoS detection using deep learning,” *Procedia Computer Science*, vol. 218, pp. 2420–2429, 2023, doi: 10.1016/j.procs.2023.01.217.

BIOGRAPHIES OF AUTHORS

Bharathi Ramachandra received the Bachelor of Engineering is Electronics and Communication Engineering from SJCE in the year 2008. Received Master degree on Computer Network Engineering from NIE Mysuru in the year 2013. Curenty working as a assistent professor in the Department of Electronics and Communication Engineering, GSSS Institute of Engineering and Technology for Women, Mysuru. Her area of intreasts are WSN, attacks in WSN, cryptography, IoT, and communication systems. She can be contacted at email: bharathi.08r@gmail.com.

T. P. Surekha Professor and Dean (student welfare), Department of Electronics and Communication Engineering, Vidyavardhaka College of Engineering, Mysuru, Karnataka, India. She has completed her Ph.D. in Communication Systems from Visvesvaraya Technological University, Belagavi, Karnataka, India. She has more than 30+ years of teaching experiemce. She has published 32 national/international journals. Her areas of intreasts are wire and wireless communication systems, bio-medical signal processing, and engineering education. She can be contacted at email: drtps@vvce.ac.in.