IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 15, No. 2, April 2026, pp. 1505~1512
ISSN: 2252-8938, DOI: 10.11591/ijai.v15.i2.pp1505-1512

Zoneout regularization-gated recurrent unit algorithm on NIDS with class imbalance handling

Mala Kariyappa[1], Manjunath Hanumanthappa Rangappa[2], Venugopal Dasappa[3], Gururaja Hebbur Satyanarayana[4], Girish Keshava Rao[5], Gousia Thahniyath[6]
[1]Department of Information Science and Engineering, Channabaseveshwara Institute of Technology, Tumkur, India
[2]Department of Artificial Intelligence and Machine Learning, Jyothy Institute of Technology, Bengaluru, India
[3]Department of Computer Science and Engineering, Harsha Institute of Technology, Bengaluru, India
[4]Department of Computer Science and Business Systems, B. M. S. College of Engineering, Bengaluru, India
[5]Department of Computer Applications, B. M. S. College of Engineering, Bengaluru, India
[6]Department of Computer Science and Engineering, Dayananda Sagar University, Bengaluru, India

Article Info

Article history:

Received May 12, 2025
Revised Jan 16, 2026
Accepted Jan 25, 2026

Keywords:

Gated recurrent unit
Near Miss
Network intrusion detection system
SMOTE
Zoneout regularization

ABSTRACT

Network intrusion detection system (NIDS) is primarily utilized tool to identify malicious threats on the network. It plays an essential role in safeguarding against an increasing variety of attacks and ensures enhanced security for the network. The existing model struggled to handle the imbalance of class issues during the process of classification due to their biased nature, which reduced the performance of the algorithm. In this paper, the zoneout regularization–gated recurrent unit (ZR-GRU) algorithm is developed to detect and classify intrusions in the network. Incorporating the ZR into GRU reduces overfitting by preventing the model from becoming overly dependent on specific features. It provides good generalization by maintaining diversity in learned representation. Synthetic minority oversampling technique (SMOTE) and Near Miss methods are utilized to balance the samples in the dataset, which helps to increase the performance of a classifier in NIDS. The ZR-GRU technique attained 99.91% accuracy on UNSW-NB15, 99.92% accuracy on CIC-IDS2018, and 99.14% accuracy on CIC-DDoS2019 when comparing with a convolutional neural network-bidirectional long short-term memory (CNN-BiLSTM).

Corresponding Author:

Mala Kariyappa
Department of Information Science and Engineering, Channabaseveshwara Institute of Technology
Tumkur, India
Email: mala.k@cittumkur.org

1. INTRODUCTION

As amount of users and dependency on the internet continue to grow, network attacks have become much more frequent, disrupting standard processes [1]. The rise in various types and scales of attacks has exposed the drawbacks of traditional security measures like antivirus software, policy of security, and firewalls, which are insufficient to fully protect the networks [2], [3]. The network intrusion detection system (NIDS) is developed to prevent targeted attacks through continuously monitoring traffic in the network [4]. That system analyzes data and tries to recognize anomalies in a network, like unauthorized access, damage, or intrusions [5]. Intrusion detection system (IDS) is generally assigned as a software application for monitoring a network and activities of a system for malicious behaviors [6]. It identifies unauthorized access to a system and then reports that to network administrators [7]. Different machine learning (ML)-based algorithms have been developed for IDS for identifying and classifying security attacks [8]. The ML-based

approaches in network traffic and associated them with different pre-defined network traffic profiles and support in detecting if the specific network traffic instances are intrusions or not [9], [10]. These algorithms use ML for developing an IDS, employing either one classifier or a hybrid classifier [11]. Single classifier-based systems utilize one ML-based approaches for identifying the attacks, while hybrid systems integrate multiple ML-based algorithms, with one classifier handling the pre-processing information and the other for training the pre-processed information [12], [13]. To meet the demands of large-scale networks, the deep learning (DL)-based algorithm, which is a branch of ML is employed in NIDS [14]. Anley *et al.* [15] implemented an innovative technique to distributed denial of service (DDoS) detection using a convolutional neural network (CNN) and adaptive structures of transfer learning methods.

Zhao *et al.* [16] introduced a DDoS attack detection technique, which integrated the self-attention mechanism with CNN-bidirectional long short-term memory (BiLSTM) to address problems on high dimensionality, multi-feature dimensions, low classification accuracy, and high false positive rate in original traffic information. Zhao *et al.* [17] presented a hybrid IDS that depended on the correlation-based feature selection with differential evolution (CFS-DE) feature selection technique and the weighted stacking classification technique. The presented method minimized high-dimensional features and maximized classification performance. Yin *et al.* [18] suggested a hybrid feature selection technique on multi-class network anomalies by utilizing a multi-layer perceptron (MLP) network. Ayantayo *et al.* [19] developed three various DL-based algorithms known as early-fusion, late-fusion, and late-ensemble learning techniques utilizing feature fusion with completely integrated deep networks. Existing algorithms for IDS suffered from vanishing gradients, overfitting, and poor handling of imbalanced datasets. Recurrent neural network (RNN)-based algorithms failed to capture long-term dependencies because of gradient problems, while overfitting limits their generalization on unseen attacks. For mitigating these issues, this research explores the effectiveness of zoneout regularization–gated recurrent unit (ZR-GRU) method is developed to enhance network intrusion detection efficiently. Zoneout is particularly selected over other regularizations like dropout and batch normalization because of its temporal consistency-preserving nature, which is suitable for GRU used in NIDS. Additionally, like the synthetic minority oversampling technique (SMOTE) and Near Miss based class imbalance handling techniques are integrated with the proposed intrusion detection model the significant contributions of the article are described as follows: SMOTE and Near Miss methods are used in a pre-processing phase to balance samples in the dataset, which helps to improve the classification performance. The ZR-GRU algorithm is developed for the detection and classification of intrusions in a network, which effectively classifies the intrusions in the network with classification accuracy.

This research paper is systematized as follows. Section 2 provides details of a proposed algorithm. Section 3 gives the results and discussion of the proposed algorithm. The conclusion is given in section 4.

2. PROPOSED METHOD

This research proposed a DL-based algorithm with class balancing algorithms to detect and classify intrusions in the network. Figure 1 represents the process of NIDS. The three benchmark datasets are used in this article, which are then pre-processed by using SMOTE, Near Miss, and standardization techniques to enhance the quality of data. Then, the features are classified by using the developed ZR-GRU, which classified the intrusions effectively.

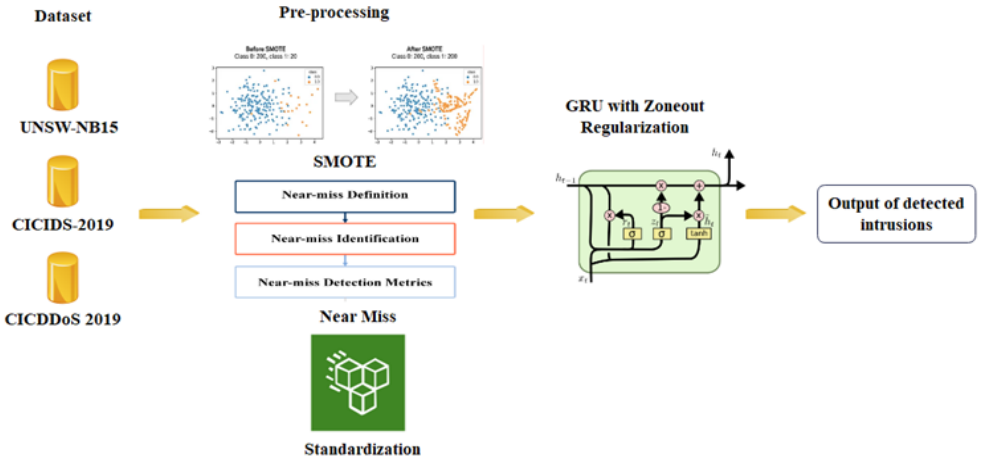

Figure 1. Process of NIDS

2.1. Dataset

The datasets used in this research are UNSW-NB15 [20], CSE-CIC-IDS2018 [21], and CIC-DDoS2019 [22]. UNSW-NB15 dataset includes nine specific attack types and 49 features. These attack types cover wide range of cyber threats and enable the complete evaluation of IDS. The UNSW-NB15 consists of 10 traffic types such as worms, shellcode, backdoor, analysis, fuzzers, reconnaissance, DoS, exploits, generic, and benign. Next, the CIC-IDS2018 dataset includes 8 types of traffic data, such as DDoS attacks-slow HTTP test, DDoS attacks-LOIC-UDP, DDoS attacks Slowloris, DDoS attacks GoldenEye, DDoS attacks-HULK, DDoS attacks-HOIC, DDoS attacks–loic–HHTP, and Benign. The CIC-DDoS2019 dataset involves 13 various types of traffic data like WebDDoS, Portmap, NetBIOS, LDAP, Benign, DNS, SNMP, MSSQL, SYN, SSDP, NTP, UDP, and TFTP respectively.

2.2. Pre-processing

The dataset is pre-processed by using three different pre-processing algorithms like SMOTE, Near Miss method, and standardization. SMOTE and Near Miss techniques are used for effectively handling minority and majority classes in NIDS. A detailed description of these pre-processing techniques is given as follows.

2.2.1. Synthetic minority oversampling technique

SMOTE [23] is a resampling algorithm that maximizes the quantity of minority class samples by developing synthetic samples in the minority class and is employed to balance data with a high unbalanced ratio. The primary aim of SMOTE is to produce new samples in the minority class data through interpolation among samples of classes that are nearer to each other. Initially, the N that is the chosen quantity of oversampling is set to an integer number. This number is selected to balance the dataset, achieving a 1:1 ratio across various classes. Next, three major phases are considered iteratively. These phases are random selection of samples that belong to the minority class. The k-nearest neighbors of the sample are chosen, and the N of these K neighbors are chosen randomly for interpolation and producing new samples.

2.2.2. Near Miss technique

This technique balances class distribution to classification datasets with skewed class distribution called as under-sampling. For balancing class distribution, under-sampling eliminates training dataset samples that pertain to the majority class like minimizing skew from a 1:100 to a 1:10, 1:2 or 1:1 class distribution. For evaluating the influence of the data-point technique, this article utilizes an under-sampling strategy depended on a Near Miss technique. Near Miss is selected depending on the benefits of providing a much more robust and fair class distribution boundary that is identified to enhance classifier performance to detect large-scale imbalanced data.

2.2.3. Standardization

The standardization [24] of data removes the influence of varying scales of various elements on step sizes and avoids unwanted overhead. Standardization ensures 0 mean and 1 standard deviation, providing an advantage for statistical analysis. The mathematical equation for standardization is given in (1). In (1), X is the actual data, μ is the mean, σ is the standard deviation, and X_{new} is standardized data.

$$X_{new} = \frac{X-\mu}{\sigma} \quad (1)$$

2.3. Classification

GRU is advanced version of RNN with a gating mechanism and closer to the standard LSTM network, but with fewer training parameters. GRU is a very suitable RNN version for short sequences of input data [25]. GRU involves two gates such as update gate which supports to determine how much information is forwarded based on past time steps. The reset gate aims to identify the amount of data that is removed from the past.

Additionally, the GRU detaches the cell state, and data is transferred through a hidden state. The parameters considered in this manuscript for GRU are 20 Epochs, sparse categorical cross entropy loss function, 32 batch size, Adam optimizer, and SoftMax activation function. This research explores the effectiveness of ZR in the GRU method integrated with class imbalance handling techniques like SMOTE and Near Miss over multiple datasets. Zoneout is particularly selected over other regularizations like dropout and batch normalization because of its temporal consistency-preserving nature, which is suitable for GRU used in NIDS. The elements presented in the GRU architecture are described as follows. The external input data represented as E_{t_s} at time step t_s. The hidden state is represented as h_{s-1} at the past time step t_{s-1}. The reset gate manages the amount of previous data, which is forgotten after the point-by-point multiplication through h_{s-1}. The result of the update gate with the sigmoid function O_{t_s} identifies the amount of data of the previous hidden layer, which is processed in the consecutive hidden state. The candidate activation vector C_{t_s}, which utilizes the reset gate vector rg_{t_s} for conserving previous significant data. The consecutive hidden

state value h_s is similar to the outcome O_{t_s}. Every candidate activation vector and internal gate in the GRU cell for every new input data at time t_s computes certain significant updates, given in (2) to (4).

$$O_{t_s} = \sigma\left(P_o E_{t_s} + Q_o h_{s-1} + a_o\right) \tag{2}$$

$$rg_{t_s} = \sigma\left(P_{rg} E_{t_s} + Q_{rg} h_{s-1} + a_{rg}\right) \tag{3}$$

$$C_{t_s} = tanh\left(P_C E_{t_s} + Q_C\left(rg_{t_s} \otimes h_{s-1}\right) + a_c\right) \tag{4}$$

Where $P_o E_{t_s}$ represents the present input utilizing weight matrix P_o, $Q_o h_{s-1}$ represents past hidden state with weight matrix Q_o, a_o represents the bias term for result activation, and σ represents the sigmoid activation function, which ensures value between 0 and 1. $P_{rg} E_{t_s}$ represents the present input to the reset gate, $Q_{rg} h_{s-1}$ represents previous hidden state to a reset gate, and a_{rg} is bias term of a reset gate. $P_C E_{t_s}$ represents the present input of the candidate state, $Q_C\left(rg_{t_s} \otimes h_{s-1}\right)$ represents the previous hidden state of the candidate state, rg_{t_s} represents reset gate that handles how much of past hidden state is utilized, $\otimes$ represents element-wise multiplication, a_c represents bias term of a candidate hidden state, and $tanh$ is activation function. Through manipulating a vector of the update gate, the further hidden layer and the last outcome of the GRU cell unit are attained, and its mathematical formula is given as (5). In (5), h_s represents the hidden state at time s, O_{t_s} represents output gate at time s, C_{t_s} represents the candidate hidden state, $\left(1 - O_{t_s}\right)$ represents the complement of the output gate, and $\otimes$ represents element-wise multiplication.

$$h_s = O_{t_s} \otimes C_{t_s} + \left(1 - O_{t_s}\right) \otimes h_{s-1} \tag{5}$$

2.3.1. Zoneout regularization

Zoneout is particularly selected over other regularizations like dropout and batch normalization because of its temporal consistency-preserving nature, which is suitable for GRU used in NIDS. Unlike dropout that randomly deactivates neurons and disrupts sequence continuity. ZR randomly preserves past hidden states and handles sequential integrity, which is essential in network traffic data.

Batch normalization is efficient in feedforward networks but underperforms in RNNs due to difficulties in employing consistent normalization over time steps. ZR outperforms dropout-GRU and conventional GRU methods. In every timestep, zoneout stochastically forces certain hidden units to handle their past scores. Unlike dropout, zoneout utilizes random noise for training a pseudo-ensemble and improves generalization ability. Through protecting rather than dropping hidden units, gradients, and state data are quickly propagated over time in feedforward stochastic depth networks. By incorporating ZR into GRU, minimizes overfitting of the model by relying on some features and provides good generalization by maintaining diversity in representation.

3. RESULTS AND DISCUSSION

The performance of the ZR-GRU algorithm is simulated in a Python environment, and the required system configurations are an i5 processor, 8 GB RAM, and Windows 10 (64-bit). Dataset samples are split in an 80:20 for training and testing sets. Evaluation metrics like accuracy, recall, F1-score, and precision are taken to analyze the performance of the ZR-GRU algorithm.

In Table 1, the performance of SMOTE+Near Miss balancing technique is evaluated using three benchmark datasets. Random under sampling, AdaSyn, traditional Near Miss, and SMOTE are taken to evaluate a performance of SMOTE+Near Miss balancing technique. SMOTE+Near Miss technique attained high accuracy on UNSW-NB15, CIC-IDS2018, and CIC-DDoS2019 datasets respectively.

In Table 2, the performance of the GRU with zoneout technique is evaluated using UNSW-NB15, CIC-IDS2018, and CIC-DDoS2019 with different metrics. The traditional GRU, GRU+dropout, GRU+L2, and GRU+L1 are taken to evaluate the performance of GRU with zoneout technique. In Table 3, the performance of ZR-GRU technique is evaluated using three various intrusion datasets. The RNN, LSTM, Bi-LSTM, and traditional GRU are taken to evaluate the performance of the ZR-GRU technique.

3.1. Comparative analysis

Performance of ZR-GRU technique is compared to existing techniques like adaptive transfer learning [16], CNN-BiLSTM [17], hybrid framework with CFS-DE [18], IGRF-RFE+MLP [19], and deep fusion mechanism [20]. The ZR-GRU technique attained high detection accuracy on UNSW-NB15, CIC-IDS2018, and CIC-DDoS2019 datasets when compared to existing traditional detection models. Table 4 represents the comparative analysis of the proposed ZR-GRU algorithm.

3.2. Discussion

The performance and results of a proposed algorithm are evaluated with three datasets using different class balancing techniques and classifiers. Moreover, the performance of a developed technique is compared with other existing techniques, like adaptive transfer learning [16], CNN-BiLSTM [17], hybrid framework with CFS-DE [18], IGRF-RFE+MLP [19], and deep fusion mechanism [20]. The existing model struggled to handle the class imbalance issue during the process of classification due to their biased nature, which reduced the performance of the models. In this paper, the ZR-GRU algorithm is developed to detect and classify intrusions in the network. Incorporating ZR in GRU minimizes overfitting by preventing the model from over-relying on specific features while providing good generalization by preserving diversity in the learned representation. SMOTE and Near Miss methods are utilized to balance the samples in the dataset, which helps improve classifier performance in NIDS.

Table 1. Performance of SMOTE+Near Miss balancing technique

Datasets	Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
UNSW-NB15	Near Miss	93.81	89.55	89.94	89.7
	Random under sampling	95.21	91.35	91.84	91.7
	AdaSyn	97.21	92.65	93.24	92.8
	SMOTE	98.61	93.65	94.84	93.9
	SMOTE+Near Miss	99.91	95.45	96.24	95.8
CIC-IDS2018	Near Miss	92.32	94.23	93.62	94.72
	Random under sampling	94.02	95.33	95.42	96.02
	AdaSyn	96.02	97.33	96.92	97.12
	SMOTE	98.02	98.43	98.82	98.42
	SMOTE+Near Miss	99.92	99.93	99.92	99.92
CIC-DDoS2019	Near Miss	92.74	93.62	94.58	93.4
	Random under sampling	94.54	95.52	95.68	95.1
	AdaSyn	96.14	96.52	97.48	96.8
	SMOTE	97.24	98.02	98.68	97.8
	SMOTE+Near Miss	99.14	99.72	99.68	99.7

Table 2. Performance of different regularization algorithms

Datasets	Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
UNSW-NB15	GRU	92.31	90.65	91.02	90.81
	GRU+dropout	94.02	91.34	91.92	91.62
	GRU+L2	95.27	92.12	92.65	92.37
	GRU+L1	94.75	91.54	91.92	91.64
	GRU with zoneout	99.91	95.45	96.24	95.80
CIC-IDS2018	GRU	95.74	95.68	95.92	95.59
	GRU+dropout	96.83	96.34	96.91	96.51
	GRU+L2	97.65	97.43	97.88	97.54
	GRU+L1	97.21	97.02	97.56	97.08
	GRU with zoneout	99.92	99.93	99.92	99.92
CIC-DDoS2019	GRU	94.83	94.21	94.56	94.36
	GRU+Dropout	96.18	96.12	96.45	96.19
	GRU+L2	96.92	96.71	96.85	96.74
	GRU+L1	96.54	96.02	96.44	96.10
	GRU with zoneout	99.14	99.72	99.68	99.70

Table 3. Performance of ZR-GRU classifier

Datasets	Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
UNSW-NB15	RNN	97.31	92.05	92.64	93.00
	LSTM	97.81	92.95	93.64	93.50
	BiLSTM	98.31	93.85	94.64	94.00
	GRU	98.91	94.75	95.24	94.90
	ZR-GRU	99.91	95.45	96.24	95.80
CICIDS 2018	RNN	97.22	96.43	97.32	96.42
	LSTM	97.72	97.23	97.92	97.32
	BiLSTM	98.62	98.13	98.72	98.32
	GRU	99.32	98.93	99.22	99.22
	ZR-GRU	99.92	99.93	99.92	99.92
CIC-DDoS 2019	RNN	98.91	94.75	95.24	94.90
	LSTM	97.24	97.72	97.38	97.50
	BiLSTM	97.94	98.22	98.38	98.20
	GRU	98.54	99.12	99.18	98.70
	ZR-GRU	99.14	99.72	99.68	99.70

Table 4. Comparative analysis of ZR-GRU algorithm

Datasets	Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
UNSW-NB15	Adaptive transfer learning [16]	99.84	NA	NA	NA
	IGRF-RFE+MLP [19]	84.24	NA	NA	82.85
	Deep fusion mechanism [20]	77.84	86.04	69.50	NA
	Proposed ZR-GRU algorithm	99.91	95.45	96.24	95.8
CIC-DDOS2019	Adaptive transfer learning [16]	93.62	NA	NA	NA
	CNN-BiLSTM [17]	95.67	95.82	95.90	95.86
	Proposed ZR-GRU algorithm	99.14	99.72	99.68	99.70
CICIDS2018	Adaptive transfer learning [16]	99.92	NA	NA	NA
	Hybrid framework with CFS-DE [18]	99.87	99.88	99.87	99.88
	Proposed ZR-GRU algorithm	99.92	99.93	99.92	99.92

4. CONCLUSION

This research introduced class balancing techniques and a DL-based algorithm for detecting network intrusions and obtaining high accuracy. Initially, the classes were balanced using SMOTE and Near Miss techniques, which balanced the samples in the dataset. Then, the balanced classes are standardized using the standardization technique. Next, the standardized features were classified using the ZR-GRU technique with high classification accuracy. Incorporating ZR in GRU minimizes overfitting by relying on some features and provides good generalization by maintaining diversity in representation. The ZR-GRU algorithm minimizes overfitting and the gradient vanishing issue during the training process that helps to enhance the classification performance of NIDS. The ZR-GRU technique attained 99.91%, 99.92%, and 99.14% accuracy on UNSW-NB15, CIC-IDS2018, and CIC-DDoS2019 datasets. In the future, a feature selection phase will be used to reduce irrelevant and inappropriate features for enhancing classification performance.

ACKNOWLEDGMENTS

The authors would like to express their sincere thanks to the Department of Information Science and Engineering, Channabaseveshwara Institute of Technology, Tumkur, and the Department of Information Science and Engineering, Sri Siddhartha Institute of Technology, Tumkur, for providing the necessary support and facilities to carry out this research work.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Mala Kariyappa	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓		✓	✓
Manjunath Hanumanthappa Rangappa		✓		✓		✓		✓	✓	✓	✓	✓		
Venugopal Dasappa		✓		✓		✓		✓		✓		✓	✓	
Gururaja Hebbur Satyanarayana	✓		✓		✓	✓	✓	✓	✓				✓	
Girish Keshava Rao		✓		✓		✓		✓	✓	✓	✓	✓		
Gousia Thahniyath			✓	✓	✓		✓	✓		✓	✓	✓	✓	

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The datasets generated during the current study are available in UNSW Sydney at https://research.unsw.edu.au/projects/unsw-nb15-dataset, reference [20]; University of New Brunswick at https://www.unb.ca/cic/datasets/ids-2018.html, reference [21] and https://www.unb.ca/cic/datasets/ddos-2019.html, reference [22].

REFERENCES

[1] N. Thockchom, M. M. Singh, and U. Nandi, "A novel ensemble learning-based model for network intrusion detection," *Complex and Intelligent Systems*, vol. 9, no. 5, pp. 5693–5714, Apr. 2023, doi: 10.1007/s40747-023-01013-7.

[2] A. Abdelkhalek and M. Mashaly, "Addressing the class imbalance problem in network intrusion detection systems using data resampling and deep learning," *Journal of Supercomputing*, vol. 79, no. 10, pp. 10611–10644, Feb. 2023, doi: 10.1007/s11227-023-05073-x.

[3] A. K. Mananayaka and S. S. Chung, "Network intrusion detection with two-phased hybrid ensemble learning and automatic feature selection," *IEEE Access*, vol. 11, pp. 45154–45167, May 2023, doi: 10.1109/ACCESS.2023.3274474.

[4] Y. Xie and H. Chen, "A novel method for effective intrusion detection based on convolutional speaking neural networks," *Journal of King Saud University - Computer and Information Sciences*, vol. 36, no. 2, Feb. 2024, doi: 10.1016/j.jksuci.2024.101975.

[5] M. I. T. Hussan, G. V. Reddy, P. T. Anitha, A. Kanagaraj, and P. Naresh, "DDoS attack detection in IoT environment using optimized Elman recurrent neural networks based on chaotic bacterial colony optimization," *Cluster Computing*, vol. 27, no. 4, pp. 4469–4490, Nov. 2024, doi: 10.1007/s10586-023-04187-4.

[6] S. Sivamohan and S. S. Sridhar, "An optimized model for network intrusion detection systems in industry 4. 0 using XAI based Bi-LSTM framework," *Neural Computing and Applications*, vol. 35, no. 15, pp. 11459–11475, Mar. 2023, doi: 10.1007/s00521-023-08319-0.

[7] I. O.-Fernandez, M. Sestelo, J. C. Burguillo, and C. P.-Blanco, "Network intrusion detection system for DDoS attacks in ICS using deep autoencoders," *Wireless Networks*, vol. 30, no. 6, pp. 5059–5075, Jan. 2024, doi: 10.1007/s11276-022-03214-3.

[8] A. V. Turukmane and R. Devendiran, "M-MultiSVM: an efficient feature selection assisted network intrusion detection system using machine learning," *Computers and Security*, vol. 137, Feb. 2024, doi: 10.1016/j.cose.2023.103587.

[9] B. Brenner *et al.*, "Better safe than sorry: risk management based on a safety-augmented network intrusion detection system," *IEEE Open Journal of the Industrial Electronics Society*, vol. 4, pp. 287–303, Jul. 2023, doi: 10.1109/OJIES.2023.3297057.

[10] S. Layeghy, M. Baktashmotlagh, and M. Portmann, "DI-NIDS: domain invariant network intrusion detection system," *Knowledge-Based Systems*, vol. 273, Aug. 2023, doi: 10.1016/j.knosys.2023.110626.

[11] R. Devendiran and A. V. Turukmane, "Dugat-LSTM: deep learning based network intrusion detection system using chaotic optimization strategy," *Expert Systems with Applications*, vol. 245, Jul. 2024, doi: 10.1016/j.eswa.2023.123027.

[12] A. Muhammad, I. Murtza, A. Saadia, and K. Kifayat, "Cortex-inspired ensemble-based network intrusion detection system," *Neural Computing and Applications*, vol. 35, no. 21, pp. 15415–15428, Apr. 2023, doi: 10.1007/s00521-023-08561-6.

[13] Y. G. Damtew, H. Chen, and Z. Yuan, "Heterogeneous ensemble feature selection for network intrusion detection system," *International Journal of Computational Intelligence Systems*, vol. 16, no. 1, Feb. 2023, doi: 10.1007/s44196-022-00174-6.

[14] A. Shiravani, M. H. Sadreddini, and H. N. Nahook, "Network intrusion detection using data dimensions reduction techniques," *Journal of Big Data*, vol. 10, no. 1, Mar. 2023, doi: 10.1186/s40537-023-00697-5.

[15] M. B. Anley, A. Genovese, D. Agostinello, and V. Piuri, "Robust DDoS attack detection with adaptive transfer learning," *Computers and Security*, vol. 144, Sep. 2024, doi: 10.1016/j.cose.2024.103962.

[16] J. Zhao, Y. Liu, Q. Zhang, and X. Zheng, "CNN-AttBiLSTM mechanism: a DDoS attack detection method based on attention mechanism and CNN-BiLSTM," *IEEE Access*, vol. 11, pp. 136308–136317, Nov. 2023, doi: 10.1109/ACCESS.2023.3334916.

[17] R. Zhao, Y. Mu, L. Zou, and X. Wen, "A hybrid intrusion detection system based on feature selection and weighted stacking classifier," *IEEE Access*, vol. 10, pp. 71414–71426, Jun. 2022, doi: 10.1109/ACCESS.2022.3186975.

[18] Y. Yin *et al.*, "IGRF-RFE: a hybrid feature selection method for MLP-based network intrusion detection on UNSW-NB15 dataset," *Journal of Big Data*, vol. 10, no. 1, Feb. 2023, doi: 10.1186/s40537-023-00694-8.

[19] A. Ayantayo *et al.*, "Network intrusion detection using feature fusion with deep learning," *Journal of Big Data*, vol. 10, no. 1, Nov. 2023, doi: 10.1186/s40537-023-00834-0.

[20] N. Moustafa, "The UNSW-NB15 dataset," *UNSW Sydney*. Accessed: Dec. 15, 2025. [Online]. Available: https://research.unsw.edu.au/projects/unsw-nb15-dataset

[21] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "CSE-CIC-IDS2018 on AWS," *University of New Brunswick.* Accessed: Dec. 15, 2025. [Online]. Available: https://www.unb.ca/cic/datasets/ids-2018.html

[22] M. Rabbani, "DDoS evaluation dataset (CIC-DDoS2019)," *University of New Brunswick.* Accessed: Dec. 15, 2025. [Online]. Available: https://www.unb.ca/cic/datasets/ddos-2019.html

[23] Z. Allal, H. N. Noura, O. Salman, and K. Chahine, "Leveraging the power of machine learning and data balancing techniques to evaluate stability in smart grids," *Engineering Applications of Artificial Intelligence*, vol. 133, Jul. 2024, doi: 10.1016/j.engappai.2024.108304.

[24] A. Petersen, M. Brabrand, and S. Kucheryavskiy, "Using artificial neural networks for anomaly detection in infrared thermography images for rapid diagnosis in an emergency care unit," *Security and Communication Networks*, vol. 112, Feb. 2026, doi: 10.1016/j.bspc.2025.108734.

[25] H. Ma, J. Cao, B. Mi, D. Huang, Y. Liu, and S. Li, "A GRU-based lightweight system for CAN intrusion detection in real time," *Security and Communication Networks*, vol. 2022, pp. 1–11, Jun. 2022, doi: 10.1155/2022/5827056.

BIOGRAPHIES OF AUTHORS

Mala Kariyappa received her B.E. degree in Computer Science and Engineering from SJB Institute of Technology, Bengaluru, India, in 2014, and completed her M.Tech. in Computer Science and Engineering from Sri Siddhartha Institute of Technology, Tumkur, India, in 2018. She has been serving as an assistant professor in the Department of Information Science and Engineering since 2019. She received her Ph.D. from Sri Siddhartha Institute of Technology. She can be contacted at email: mala.k@cittumkur.org.

Manjunath Hanumanthappa Rangappa received his B.E. degree in Computer Science and Engineering from S.J.M. Institute of Technology, Chitradurga, affiliated with Kuvempu University, India, and his M.Tech. degree in Computer Science and Engineering from Basaveshwara Engineering College, Bagalkot, affiliated with Visvesvaraya Technological University. He obtained his Ph.D. degree from Sri Siddhartha Academy of Higher Education (Sri Siddhartha University), Tumkur, India, in 2022. He is currently serving as an associate professor in the Department of Artificial Intelligence and Machine Learning. He is a life member of ISTE and IETE. He can be contacted at email: manjunath.hr@jyothyit.ac.in.

Venugopal Dasappa received his B.E. degree in Computer Science and Engineering in 2010 and his M.Tech. degree in Computer Science and Engineering in 2012 from Channabasaveshwara Institute of Technology, Gubbi, affiliated with Visvesvaraya Technological University, Belagavi, India. He is presently working as an assistant professor at Harsha Institute of Technology, Nelamangala, Bengaluru. Previously, he served as an assistant professor at Siddaganga Institute of Technology, Tumakuru, and at Akshaya Institute of Technology, Tumakuru. He also has industry experience as a software developer at One67 Digital Solutions Pvt. Ltd. He can be contacted at email: venugd89@gmail.com.

Gururaja Hebbur Satyanarayana received his doctoral degree in Computer Science and Engineering and is currently serving as an assistant professor (selection grade) in the Department of Computer Science and Business Systems at B. M. S. College of Engineering, Bengaluru, India. His research interests include computer science applications, information systems, and emerging computing technologies. He has published research articles in international journals and conferences indexed in Scopus. He can be contacted at email: gururajhs.ise@bmsce.ac.in.

Girish Keshava Rao is currently serving as an assistant professor in the Department of Computer Applications at B. M. S. College of Engineering, Bengaluru, India. His academic and research interests include computer applications, data analytics, and emerging computing technologies. He has published research articles in international journals and conferences. He can be contacted at email: gk.mca@bmsce.ac.in.

Gousia Thahniyath received her B.E. degree in Computer Science and Engineering from Gulbarga University and her M.E. degree from Bangalore University. She obtained her Ph.D. degree in Computer Science and Engineering from Visvesvaraya Technological University, with doctoral research focused on intrusion detection systems for wireless sensor networks. She is currently serving as an associate professor in the Department of Computer Science and Engineering at Dayananda Sagar University, Bengaluru, India. She has published extensively in SCI- and Scopus-indexed journals, conferences, and book chapters and has received funded research projects from recognized agencies. She can be contacted at email: gousia-cse@dsu.edu.in.