The Zoneout Regularized Gated Recurrent Unit Algorithm for Network Intrusion Detection with Class Imbalance Mitigation

K. Mala
Department of Computer Science and Engineering, Sri Siddhartha Institute of Technology, Sri, India | Siddhartha Academy of Higher Education, Tumakuru, India
mala.k@cittumkur.org (corresponding author)

H. S. Annapurna
Department of Information Science and Engineering, Sri Siddhartha Institute of Technology, Sri, India | Siddhartha Academy of Higher Education, Tumakuru, India
annapooranahs@ssit.edu.in

Received: 5 May 2025 | Revised: 12 May 2025 and 27 May 2025 | Accepted: 31 May 2025

 DOI: https://doi.org/10.48084/etasr.11823

ABSTRACT

This study used a Zoneout-Regularized Gated Recurrent Unit (ZR-GRU) to enhance the effectiveness of Network Intrusion Detection Systems (NIDSs) by maintaining long-term temporal patterns and reducing the risk of overfitting. In contrast to traditional models, ZR-GRU incorporates zoneout regularization to improve generalization over diverse network traffic patterns. To address the prevalent issue of class imbalance in network security datasets, the model integrates the Synthetic Minority Oversampling Technique (SMOTE) for oversampling minority classes and NearMiss for undersampling majority classes, promoting balanced class representation. The model was evaluated on three widely used benchmark datasets, UNSW-NB15, CICIDS 2018, and CIC-DDoS 2019, chosen due to their realistic network traffic characteristics and the diverse range of contemporary attack types. ZR-GRU achieved high accuracy rates of 99.91%, 99.92%, and 99.14% on these datasets, outperforming traditional architectures. The findings highlight the strength, flexibility, and effectiveness of the model for real-time and adaptive intrusion detection in diverse network settings.

Keywords-gated recurrent unit; network intrusion detection system; near miss; synthetic minority oversampling technique; zoneout regularization

I. INTRODUCTION

The growing reliance on Internet-connected systems has significantly increased exposure to cyber threats, highlighting the shortcomings of conventional security measures, such as firewalls and antivirus programs. This has elevated the importance of NIDSs, which play a crucial role in identifying anomalous network behavior in real time. Although Machine Learning (ML) techniques have contributed to the advancement of intrusion detection, persistent issues. such as overfitting, high-dimensional feature spaces, and imbalanced data, continue to limit performance [1, 2]. Although ensemble and hybrid approaches can enhance detection accuracy, they often struggle with scalability and effectively addressing data imbalance [1, 3]. Techniques involving Convolutional Neural Networks (CNNs) and Bidirectional LSTM (Bi-LSTM) are proficient at extracting spatial and temporal patterns but tend to lack cross-dataset generalization [4-6].

In [2], the class imbalance was addressed by resampling, but the deep learning model showed poor generalization. In [3], a hybrid model with feature selection was proposed, although it introduced significant complexity. In [7], a CNN-Attention-BiLSTM model was proposed for DDoS detection, but its performance was limited by dataset dependence. In [8], stacking was used to improve the results but faced problems with vanishing gradients. In [9], a hybrid feature selection was applied, which led to overfitting in multiclass tasks, while in [10], feature fusion was effective but was still affected by data imbalance. In [11], a hybrid CNN-LSTM network was proposed to identify intrusions for IoT environments using a Raspberry Pi 3, achieving 98.78% accuracy on a single dataset. These findings highlight the need for more generalized and efficient detection approaches.

This study proposes a ZR-GRU that integrates Zoneout to preserve temporal dependencies and reduce overfitting more effectively than Dropout in sequential traffic modeling. To address class imbalance, the SMOTE and NearMiss are used

during preprocessing, ensuring fair learning across the majority and minority classes. Compared to CNN-BiLSTM [7] and optimized recurrent models [5], ZR-GRU offers lower complexity and faster inference, supporting real-time applications. This study evaluates ZR-GRU on three benchmark datasets, UNSW-NB15 [12-16], CICIDS 2018 [17], and CIC-DDoS 2019 [18], which include diverse traffic behaviors and multiple categories of cyberattacks. The model consistently achieves high accuracy and generalization, proving effective against both frequent and rare threats. In summary, this work introduces a lightweight, balanced, and generalizable intrusion detection framework that efficiently overcomes key challenges, such as vanishing gradients, overfitting, and class imbalance.

II. PROBLEM STATEMENT AND PROPOSED METHOD

Traditional IDSs struggle to detect rare or sophisticated attacks due to class imbalance, poor temporal modeling, and high model complexity [2, 6, 7]. Existing deep models, such as LSTM and CNN-BiLSTM, often fail to retain long-term dependencies or generalize well across various intrusion types [4, 7]. To address these limitations, this work proposes a lightweight, regularized GRU with hybrid SMOTE and NearMiss to improve accuracy and robustness on multiclass intrusion datasets [2, 19]. The UNSW-NB15 [12-16], CSE-CIC IDS 2018 [17], and CIC DDoS 2019 [18] datasets are pre-processed with SMOTE, NearMiss, and standardization techniques to enhance data quality [2, 19, 20]. The features are then classified using the ZR-GRU [5, 6, 21], effectively identifying intrusions while preserving temporal information.

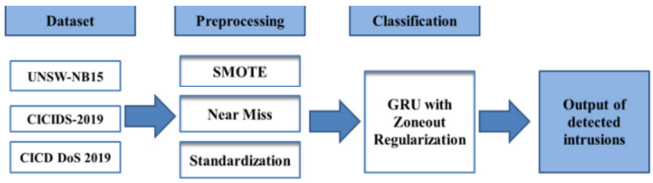

Fig. 1. Proposed NIDS process.

A. Dataset

The UNSW-NB15 dataset includes 49 features and nine distinct attack categories, such as Fuzzers, Backdoors, Exploits, DoS, Generic, Shellcode, Analysis, Worms, and Reconnaissance. This diversity makes it highly suitable to evaluate how well an intrusion detection model performs across a wide range of cyberattack scenarios [12]. The UNSW-NB15 provides recent and varied threat data through a hybrid collection approach. CICIDS2018 expands coverage to include Botnets, Web-based threats, and Brute Force attacks [17], while CIC-DDoS2019 focuses on large-scale DDoS patterns, such as SYN and UDP floods [18]. Collectively, they support comprehensive training and evaluation, ensuring that the proposed model generalizes well across different intrusion contexts.

B. Preprocessing

The dataset was preprocessed using three main techniques: SMOTE NearMiss, and standardization. SMOTE and NearMiss were applied both independently and sequentially to mitigate class imbalance [2, 19]. SMOTE improved the recall of the model by creating artificial examples for underrepresented classes, increasing its ability to identify infrequent attack instances with greater accuracy [2]. NearMiss was used as an undersampling method, enhancing precision by refining the majority class distribution and reducing false positives [19]. When combined, these techniques balance the dataset more effectively, leading to better F1 scores. The standardization was then applied to ensure that all features contributed equally during training [20].

1) Synthetic Minority Oversampling Technique

SMOTE is an oversampling technique that addresses the imbalance of the dataset by producing synthetic samples for the minority class [2, 19]. Instead of replicating data, it interpolates between existing minority instances, thus lowering the chances of overfitting. The method works by randomly selecting a minority sample, identifying its k-nearest neighbors, and generating new points along the lines connecting them. This approach refines the decision boundaries of classifiers, especially in scenarios where the class distribution is significantly skewed.

2) NearMiss

NearMiss is an undersampling technique that balances class distributions by selectively reducing majority class samples [19]. It enhances model precision by removing majority instances that are far from the decision boundary, thus minimizing false positives. NearMiss adjusts class ratios, balancing the learning process and reducing the classifier's bias toward the majority class.

3) Standardization

Standardization removes the influence of differing feature scales during training, allowing optimization algorithms to converge faster and more accurately [20]. Unlike normalization, which rescales features to a specific range, standardization transforms the data to have a mean of 0 and a standard deviation of 1, which is more suitable for neural networks and statistical analysis [20]. Standardization is also less sensitive to outliers. The mathematical equation for standardization is:

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

where X represents the actual data, μ represents the mean, σ represents the standard deviation, and X_{new} represents the standardized data.

C. Classification

The GRU is a simplified Recurrent Neural Network (RNN) that models temporal dependencies and addresses the vanishing gradient problem in traditional RNNs [5, 6]. It is more efficient than LSTM due to its fewer parameters. GRU uses two gates, the reset gate for managing past information and the update gate for retaining state. This study improves GRU with

Zoneout regularization for better temporal consistency and applies SMOTE and NearMiss to handle class imbalance in the datasets [2, 5]. Zoneout surpasses Dropout in sequential learning, making it well-suited for intrusion detection. The training process involved 20 epochs with a batch size of 32, utilizing sparse categorical cross-entropy as the loss function, the Adam optimization algorithm, and softmax activation for output classification.

In GRU, the input data at the time step t_s are denoted as E_{t_s}, while the hidden state from the previous step t_{s-1} is h_{s-1}. The reset gate controls the amount of past data discarded, while the update gate determines how much of the past hidden state is retained. The candidate activation vector C_{t_s} leverages the reset gate to retain significant information, and the final hidden state h_s combines the candidate activation and the previous state, capturing temporal dependencies essential for NIDSs. Every candidate activation vector and internal gate in the GRU cell for every new input data in time t_s is calculated using:

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

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

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

where $P_o E_{t_s}$ represents the present input by utilizing a weight matrix P_o, $Q_o h_{s-1}$ represents the past hidden state with the weight matrix Q_o, a_o represents the bias term for result activation, and σ represents the sigmoid activation function that ensures a value among 0 and 1. $P_{rg} E_{t_s}$ represents the present input to the reset gate, $Q_{rg} h_{s-1}$ represents the previous hidden state of the reset gate, and a_{rg} represents the bias term of the 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, the rg_{t_s} reset gate controls how much of the past hidden state is retained when computing the current state in the model, $\otimes$ represents element-wise multiplication, a_c represents the bias term of the candidate hidden state, and tanh is the activation function. The update gate vector modulates the final hidden state and output as:

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

where h_s represents a hidden state in time s, O_{t_s} represents the output gate in time s, C_{t_s} represents the candidate hidden state, and $\left(1 - O_{t_s}\right)$ represents the complement of the output gate.

D. Zoneout Regularization

Zoneout is integrated into GRU to enhance temporal regularization. Unlike Dropout [21], which randomly zeros out neuron activations and can disrupt temporal dependencies, Zoneout selectively retains previous hidden states. This retention preserves sequence continuity, which is crucial in time-series data, such as network traffic. Zoneout also differs from Batch Normalization [21], which stabilizes learning in feedforward networks but struggles in recurrent models due to inconsistent normalization across time steps. Compared to Dropout-GRU and standard GRU, ZR-GRU maintains better long-term dependencies, thereby reducing overfitting without compromising sequence integrity. In this study, the ZR-GRU was trained using the Adam optimizer, sparse categorical cross-entropy loss, a batch size of 32, 20 epochs, and softmax activation. This configuration ensures efficient training while preserving performance across imbalanced multiclass intrusion scenarios [5, 21].

III. EXPERIMENTAL ANALYSIS

The proposed ZR-GRU algorithm was rigorously evaluated in a Python-based environment equipped with an Intel i5 processor, 8 GB RAM, running Windows 10 (64-bit). The effectiveness of the model was evaluated using key performance indicators, namely Accuracy, Precision, Recall, and F1-score.

A. Performance with Class Balancing Techniques

To address class imbalance, a critical challenge in intrusion detection, four well-known data balancing techniques were applied: NearMiss, Random undersampling, AdaSyn, and SMOTE. The combination of SMOTE with NearMiss consistently showed the best results across the three datasets. To ensure robust evaluation, all experiments employed 10-fold cross-validation.

The results summarized in Table I indicate that the hybrid SMOTE and NearMiss approach outperformed each method used separately across all datasets and metrics. This is due to the ability of SMOTE to generate synthetic minority class samples and the effectiveness of NearMiss in reducing majority class dominance, leading to improved decision boundaries. Statistical tests confirmed that these performance gains are significant, demonstrating the hybrid method's effectiveness in handling class imbalance in network intrusion detection.

TABLE I. PERFORMANCE COMPARISON OF BALANCING TECHNIQUES

Method	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
UNSW-NB15 dataset				
NearMiss	93.81	89.55	89.94	89.7
Random undersampling	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+NearMiss	99.91	95.45	96.24	95.8
CICIDS 2018 dataset				
NearMiss	92.32	94.23	93.62	94.72
Random undersampling	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+NearMiss	99.92	99.93	99.92	99.92
CIC-DDoS 2019 dataset				
NearMiss	92.74	93.62	94.58	93.4
Random undersampling	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+NearMiss	99.14	99.72	99.68	99.7

B. Evaluation of the ZR-GRU Classifier

To validate its effectiveness, the ZR-GRU model was benchmarked against standard recurrent models, including RNN, LSTM, Bi-LSTM, and traditional GRU. Performance

evaluations across all three benchmark NIDS datasets demonstrate that ZR-GRU consistently outperforms other models, offering empirical proof of its superiority in managing overfitting and improving generalization. The integration of Zoneout enables the GRU to retain hidden state information stochastically, thereby enhancing long-term dependency learning without sacrificing training stability. The proposed model exhibited robust behavior under both class-balanced and imbalanced training scenarios. Table II presents a comprehensive comparison of the ZR-GRU versus other RNN-based models.

TABLE II. PERFORMANCE OF ZR-GRU CLASSIFIER

Method	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
UNSW-NB15 dataset				
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
CICIDS2018 dataset				
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-DDoS2019 dataset				
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

C. Visual Performance Analysis

Figures 2 and 3 represent the accuracy versus epochs and loss versus epochs graphs for the UNSW-NB15 dataset. Figures 4 and 5 represent the accuracy versus epochs and loss versus accuracy graphs for the CICIDS 2018 dataset.

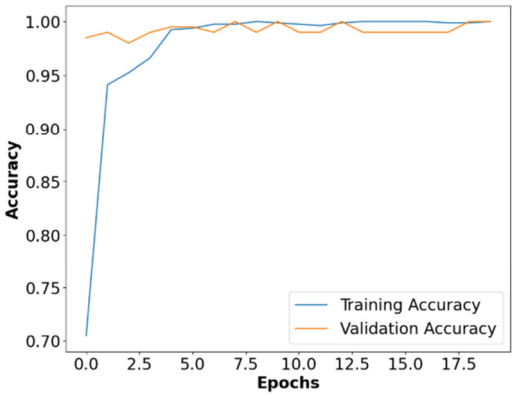

Fig. 2. Accuracy versus epochs on the UNSW-NB15 dataset.

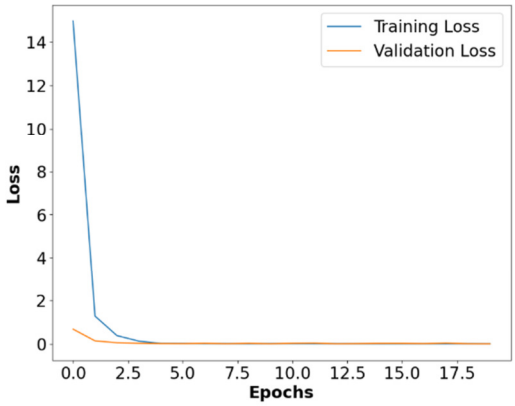

Fig. 3. Loss versus epochs on the UNSW-NB15 dataset.

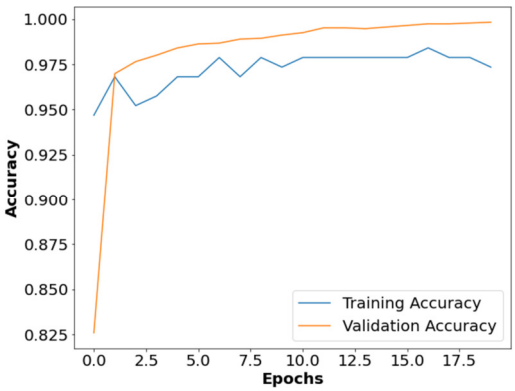

Fig. 4. Accuracy versus epochs on the CICIDS 2018 dataset.

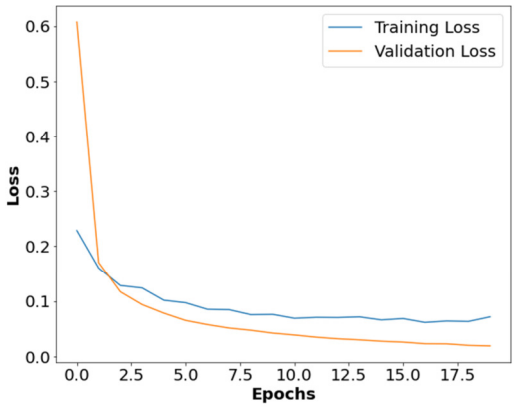

Fig. 5. Loss versus epochs on the CICIDS 2018 dataset.

Figures 6 and 7 represent the accuracy versus epochs and loss versus epochs graphs for the CIC-DDoS 2019 dataset. Figure 8 shows the training time of the proposed ZR-GRU with different epochs. The proposed algorithm was evaluated for inference time, inference speed, and GPU memory, and obtained 1.6699 s for 100 samples, 59.88 samples per s, and 1419.88 MB, respectively.

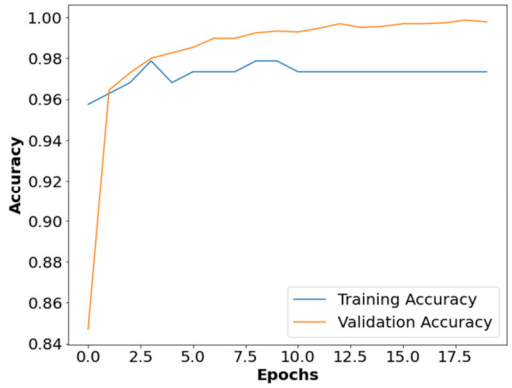

Fig. 6. Accuracy versus epochs on the CIC-DDoS 2019 dataset.

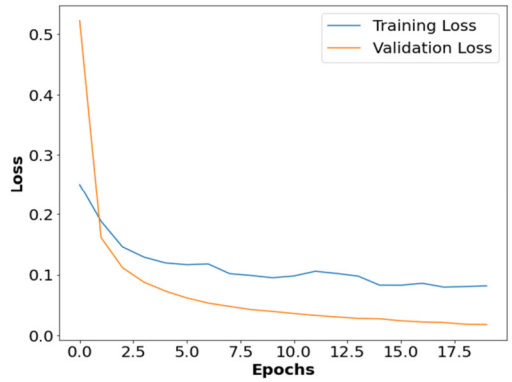

Fig. 7. Loss versus epochs on the CIC-DDoS 2019 dataset.

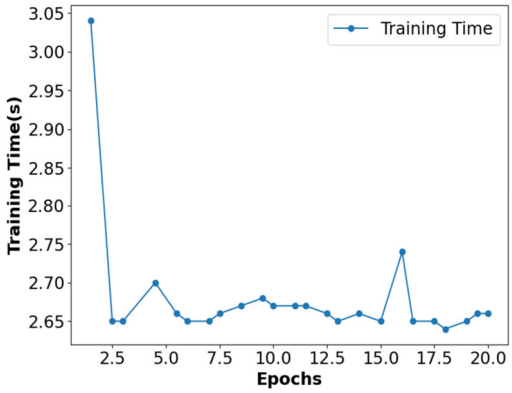

Fig. 8. Training time versus epochs.

D. Comparative Analysis

The performance of the proposed ZR-GRU model was compared with other methods, including Adaptive Transfer Learning [22], CNN-BiLSTM [7], Hybrid framework with CFS-DE [8], IGRF-RFE+MLP [9], and the Deep fusion mechanism [20] on the UNSW-NB15, CICIDS 2018, and CIC-DDoS 2019 datasets. Additionally, the proposed model was compared with GRU-based models utilizing other regularization techniques, such as Dropout and Batch Normalization. ZR-GRU consistently outperformed these approaches across all three datasets, offering strong empirical justification for the effectiveness of Zoneout regularization. These results validate that Zoneout not only enhances generalization and reduces overfitting in GRU-based architectures, but also surpasses the performance of existing hybrid and ensemble-based NIDS techniques. Table III provides a comparison of regularization techniques.

TABLE III. PERFORMANCE COMPARISON OF REGULARIZATION TECHNIQUES IN GRU-BASED MODELS

Dataset	Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
UNSW-NB15	GRU + Dropout	98.21	93.45	94.24	93.80
	GRU + Batch Normalization	98.51	94.05	94.84	94.40
	ZR-GRU (Zoneout)	99.91	95.45	96.24	95.80
CICIDS 2018	GRU + Dropout	98.82	98.03	98.42	98.12
	GRU + Batch Normalization	99.12	98.53	98.92	98.72
	ZR-GRU (Zoneout)	99.92	99.93	99.92	99.92
CIC-DDoS 2019	GRU + Dropout	97.84	97.42	97.78	97.60
	GRU + Batch Normalization	98.34	98.82	98.68	98.70
	ZR-GRU (Zoneout)	99.14	99.72	99.68	99.70

E. Discussion

The ZR-GRU architecture is designed to generalize effectively across different network environments and attack scenarios. By incorporating Zoneout regularization, it preserves temporal dependencies and reduces overfitting, making the model more adaptable to varying traffic patterns. Additionally, the integration of SMOTE and NearMiss techniques ensures balanced learning, especially for minority classes, leading to improved detection accuracy. The model achieved exceptional accuracy on UNSW-NB15 (99.91%), CICIDS 2018 (99.92%), and CIC-DDoS 2019 (99.14%), demonstrating its robustness without requiring extensive reconfiguration. In terms of computational cost, ZR-GRU outperforms more complex models, such as CNN-BiLSTM. Its streamlined architecture, aided by Zoneout regularization, enables faster inference and reduces memory usage. The experimental results showed that ZR-GRU processes 100 samples in 1.6699 s, with an inference speed of 59.88 samples per s and a GPU memory footprint of 1419.88 MB. These characteristics make ZR-GRU highly suitable for real-time deployment in resource-constrained environments, where both latency and memory efficiency are critical for effective intrusion detection.

IV. CONCLUSION

This study introduced class-balancing techniques and a DL-based algorithm to detect network intrusions and obtain high accuracy. Initially, the classes were balanced using Synthetic Minority Oversampling Technique (SMOTE) and NearMiss, and then the balanced classes were standardized. The standardized features were classified using the Zoneout-Regularized Gated Recurrent Unit (ZR-GRU) model, achieving high classification performance. Incorporating Zoneout regularization into GRU minimizes overfitting by relying on some features and provides good generalization by maintaining diversity in representation. The ZR-GRU algorithm mitigates overfitting and the gradient vanishing problem during training, improving the Network Intrusion Detection Systems (NIDSs)

classification performance. The method recorded accuracy rates of 99.91% on the UNSW-NB15 dataset, 99.92% on CICIDS 2018, and 99.14% on CIC-DDoS 2019. In future research, a feature selection stage will be introduced to filter out non-contributing features and enhance overall classification effectiveness.

REFERENCES

[1] N. Thockchom, M. M. Singh, and U. Nandi, "A novel ensemble learning-based model for network intrusion detection," *Complex & Intelligent Systems*, vol. 9, no. 5, pp. 5693–5714, Oct. 2023, https://doi.org/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," *The Journal of Supercomputing*, vol. 79, no. 10, pp. 10611–10644, Jul. 2023, https://doi.org/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, 2023, https://doi.org/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, Art. no. 101975, https://doi.org/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, Jul. 2024, https://doi.org/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, May 2023, https://doi.org/10.1007/s00521-023-08319-0.

[7] 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, 2023, https://doi.org/10.1109/ACCESS.2023.3334916.

[8] 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, 2022, https://doi.org/10.1109/ACCESS.2022.3186975.

[9] 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, Art. no. 15, https://doi.org/10.1186/s40537-023-00694-8.

[10] A. Ayantayo *et al.*, "Network intrusion detection using feature fusion with deep learning," *Journal of Big Data*, vol. 10, no. 1, Nov. 2023, Art. no. 167, https://doi.org/10.1186/s40537-023-00834-0.

[11] R. H. Altaie and H. K. Hoomod, "An Intrusion Detection System using a Hybrid Lightweight Deep Learning Algorithm," *Engineering, Technology & Applied Science Research*, vol. 14, no. 5, pp. 16740–16743, Oct. 2024, https://doi.org/10.48084/etasr.7657.

[12] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *2015 Military Communications and Information Systems Conference (MilCIS)*, Canberra, Australia, Nov. 2015, pp. 1–6, https://doi.org/10.1109/MilCIS.2015.7348942.

[13] N. Moustafa and J. and Slay, "The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set," *Information Security Journal: A Global Perspective*, vol. 25, no. 1–3, pp. 18–31, Apr. 2016, https://doi.org/10.1080/19393555.2015.1125974.

[14] N. Moustafa, J. Slay, and G. Creech, "Novel Geometric Area Analysis Technique for Anomaly Detection Using Trapezoidal Area Estimation on Large-Scale Networks," *IEEE Transactions on Big Data*, vol. 5, no. 4, pp. 481–494, Sep. 2019, https://doi.org/10.1109/TBDATA.2017.2715166.

[15] N. Moustafa, G. Creech, and J. Slay, "Big Data Analytics for Intrusion Detection System: Statistical Decision-Making Using Finite Dirichlet Mixture Models," in *Data Analytics and Decision Support for Cybersecurity: Trends, Methodologies and Applications*, I. Palomares Carrascosa, H. K. Kalutarage, and Y. Huang, Eds. Springer International Publishing, 2017, pp. 127–156.

[16] M. Sarhan, S. Layeghy, N. Moustafa, and M. Portmann, "NetFlow Datasets for Machine Learning-Based Network Intrusion Detection Systems," in *Big Data Technologies and Applications*, vol. 371, Z. Deze, H. Huang, R. Hou, S. Rho, and N. Chilamkurti, Eds. Springer International Publishing, 2021, pp. 117–135.

[17] "IDS 2018." Canadian Institute for Cybersecurity, [Online]. Available: https://www.unb.ca/cic/datasets/ids-2018.html.

[18] "DDoS 2019." Canadian Institute for Cybersecurity, [Online]. Available: https://www.unb.ca/cic/datasets/ddos-2019.html.

[19] 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, Art. no. 9, https://doi.org/10.1007/s44196-022-00174-6.

[20] 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, Art. no. 27, https://doi.org/10.1186/s40537-023-00697-5.

[21] 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, Art. no. 123027, https://doi.org/10.1016/j.eswa.2023.123027.

[22] M. B. Anley, A. Genovese, D. Agostinello, and V. Piuri, "Robust DDoS attack detection with adaptive transfer learning," *Computers & Security*, vol. 144, Sep. 2024, Art. no. 103962, https://doi.org/10.1016/j.cose.2024.103962.