Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 6, No.2, June : 2024

GLIDS-NET: GRAPH-BASED DEEP LEARNING APPROACHES FOR INTRUSION DETECTION IN FOG COMPUTING SYSTEMS

Dipti Prava Sahu Research Scholar, Biju Patnaik University of Technology / Department of Computer Science Engineering, Rourkela, Odisha, 769004, India E-mail: diptiparvasahu@gmail.com
Biswajit Tripathy Professor, Einstein College of Computer Application and Management / Master of Computer Applications, Khurda, Odisha, 752060, India E-mail: biswajit69@gmail.com
Leena Samantaray Professor & Principal, Ajay Binay Institute of Technology / Department of Electronics and Communication Engineering, Cuttack, Odisha, 753014, India E-mail: leena_sam@rediffmail.com

Abstract
Fog computing has found diverse applications across various industries such as healthcare, transportation, networking, and so on. However, these applications are attacked various intrusions, where data loss is more. So, this research aims to develop an Intrusion Detection System (IDS) for fog computing environments, specifically targeting cybersecurity applications within networked systems. The need for such a system arises from the increasing complexity and vulnerability of modern network infrastructures, where real-time threat detection is essential for ensuring data integrity and system reliability. Existing approaches problems include challenges in accurately detecting and responding to diverse types of network intrusions amidst dynamic and distributed fog computing architectures. To address these challenges, Graph Learning Based IDS Network (GLIDS-Net) methodology is proposed, beginning with the selection, and preprocessing of the UNSW-NB dataset a well- known benchmark dataset in cybersecurity research. Subsequently, a modified African Buffalo Optimization (MABO) algorithm is applied for feature selection to identify the most relevant network traffic features. The core of the proposed IDS involves leveraging a Deep Graph Correlation Network (DGCN) classifier, which is designed to capture complex graph-structured dependencies within network data and enable accurate intrusion detection.

Keywords:
Intrusion Detection System, Fog Computing, Cybersecurity, Feature Selection, Deep Learning, Graph Neural Networks, Network Security.

1. Introduction

Fog computing has found multiple applications across several industries. In transportation, fog computing facilitates real-time processing of data from connected vehicles, enabling intelligent traffic management, predictive maintenance, and enhanced safety measures. In healthcare, fog computing supports remote patient monitoring, ensuring timely data analysis and response for critical medical conditions. Within manufacturing, fog computing optimizes production processes by providing low-latency data analytics for quality control and supply chain management. Additionally, smart grids leverage fog computing to efficiently monitor and manage energy distribution, integrating renewable sources seamlessly. Fog computing is a decentralized computing infrastructure where data processing occurs closer to the data source, typically at the network edge rather than solely relying on centralized data centers. This approach aims to reduce latency and bandwidth usage by processing data locally, near the devices or sensors generating the data. Unlike traditional cloud computing, which processes data in remote data centers, fog computing distributes computing resources at various points between the data source and the cloud.

Intrusions pose significant challenges to the security and reliability of fog computing applications. Threats such as unauthorized access, data breaches, and denial-of-service attacks can compromise the integrity and confidentiality of sensitive data processed within fog networks. Intrusions not only disrupt service continuity but also raise concerns about data privacy and system vulnerability. Securing

fog computing environments requires robust intrusion detection and prevention mechanisms to promptly identify and mitigate potential threats. The rapid adoption of fog computing in diverse applications introduces security vulnerabilities due to its decentralized and distributed nature. The challenge lies in developing effective IDS tailored for fog environments to detect and respond to sophisticated cyber threats promptly. Addressing these vulnerabilities is crucial to ensuring the reliability, privacy, and integrity of fog computing applications across various domains.

Artificial Intelligence (AI)-enabled IDS are essential for fog computing environments due to their ability to detect and respond to evolving threats autonomously. Traditional rule-based IDS systems may struggle to keep pace with the dynamic nature of fog computing networks. AI-based IDS solutions can analyze vast volumes of data in real-time, detect anomalies, and adaptively learn from patterns of normal behavior, thereby enhancing the accuracy and efficiency of threat detection and response in fog computing applications. This capability is pivotal in safeguarding critical infrastructure and data integrity within fog computing ecosystems.

The rest of the paper is organized as follows: Section 2 provides a comprehensive survey of existing approaches in IDS for fog computing environments. Section 3 outlines the proposed method, detailing the utilization of GLIDS-Net technique for enhancing intrusion detection capabilities. These sections will be followed by Section 4, which presents the experimental results and performance evaluation of the proposed approach. Finally, Section 5 concludes the paper by summarizing the findings and discussing future research directions in this domain.

2. Related work

The literature review highlights several innovative approaches to intrusion detection in fog computing environments, each with unique strengths and limitations. Common drawbacks identified across methodologies include challenges in feature extraction, feature selection, grading accuracy, computation complexity, and potential performance limitations under specific conditions. These drawbacks collectively point towards significant research gaps in the development of robust and efficient IDS tailored for fog computing ecosystems. Enhancements in feature extraction, selection methodologies, and grading mechanisms are crucial to advancing the reliability, scalability, and effectiveness of IDS within fog computing environments. Addressing these gaps will be instrumental in ensuring the security and integrity of emerging IoT and fog computing applications across various domains.

In Zhao et al. (2023) [10], a lightweight intrusion detection model for the Internet of Things (IoT) with hybrid cloud-fog computing is proposed. The study addresses the need for efficient intrusion detection in resource-constrained IoT environments by leveraging a hybrid cloud-fog architecture. Despite its innovative approach, the method's drawback lies in its feature extraction capabilities. The model may encounter challenges in effectively extracting relevant features from IoT data streams, potentially impacting its detection accuracy. Ma et al. (2023) [11] presents a decision model of intrusion response based on Markov game in a fog computing environment. This approach aims to enhance the responsiveness of IDS within fog networks. However, the study's focus on response strategies highlights a potential drawback related to feature selection. The effectiveness of the intrusion response decision model could be impacted by suboptimal feature selection techniques, potentially leading to less accurate threat detection and response. Ajao and Apeh (2023) [12] propose a novel approach integrating blockchain technology with machine learning to address security vulnerabilities in fog computing within smart city sustainability. By combining blockchain and machine learning, the research aims to mitigate cybersecurity risks and ensure the long-term sustainability of smart city initiatives. This innovative integration contributes to advancing the security framework for fog computing applications in smart city ecosystems.

In Binbusayyis (2024) [13], a hybrid VGG19 and 2D-CNN approach for intrusion detection in fog-cloud environments is introduced. While the model leverages deep learning architectures for robust intrusion detection, its drawback is associated with grading issues. The grading mechanism used in the

model may exhibit limitations in accurately assessing the severity or criticality of detected intrusions, potentially leading to misclassification or inadequate response strategies. Sonawane (2024) [14] proposes enhanced feature optimization techniques for multiclass intrusion detection in IoT fog computing environments. Despite its focus on improving feature optimization, the study's drawback is related to computation complexity. The proposed optimization methods may introduce higher computational overhead, impacting real-time performance and scalability in resource-constrained fog computing environments. In Sonker et al. (2024), [15] the authors investigate fog computing-based security solutions for IoT-enabled electrical vehicles within smart grid environments. Key aspects include secure data transmission, authentication protocols, and anomaly detection to safeguard the integrity and reliability of smart grid operations. The research contributes to advancing the implementation of secure IoT-enabled systems in the context of electric vehicle integration within smart grid infrastructures.

Yao et al. (2023) [16] introduce a scalable anomaly-based IDS using generative adversarial networks (GANs) in a fog environment. While the model showcases scalability and innovation with GANs, a potential drawback lies in lower performance under certain conditions. The anomaly-based approach might exhibit reduced accuracy or higher false positive rates in complex IoT scenarios, affecting the overall effectiveness of intrusion detection. Syed et al. (2023) [17] proposes a fog-cloud-based IDS utilizing Recurrent Neural Networks (RNNs) and feature selection for IoT networks. Despite leveraging advanced neural network architectures, the study's drawback is associated with feature selection challenges. Inadequate feature selection techniques could lead to irrelevant or noisy features being incorporated into the detection system, reducing its accuracy and efficiency. Chakraborty et al. (2023) [18] develops a secure framework for IoT applications using deep learning in fog computing. The study's drawback is related to computation complexity. The deep learning-based security framework may require substantial computational resources, impacting its suitability for resource-constrained IoT and fog computing environments.

Azarkasb & Khasteh (2023) [19] explore advancing intrusion detection in fog computing with Support Vector Machines (SVMs) against XSS and SQL injection attacks. Despite the robustness of SVMs, the study's drawback is associated with grading issues. The effectiveness of the IDS in accurately assessing the severity and criticality of detected attacks may be limited, affecting the responsiveness of the system. Dhiyanesh et al. (2024) [20] present a study focused on enhancing healthcare data security in fog computing environments through the implementation of a Deep Spectral Gated Recurrent Neural Network (DSGRNN)-based IDS. By deploying this advanced neural network model, the study aims to significantly improve the security posture of healthcare data within fog computing frameworks, thereby ensuring patient confidentiality and data integrity.

3. Proposed Methodology

The developing an GLIDS-Net for fog computing involves a comprehensive research procedure starting from dataset selection and preprocessing to advanced feature selection using metaheuristic optimization techniques like MABO, followed by building and training a deep learning-based classifier such as DGCN for intrusion detection. The final prediction step enables the IDS to effectively detect and respond to network intrusions within fog computing architectures, contributing to enhanced cybersecurity and threat mitigation in modern networked environments. This research procedure integrates cutting-edge techniques from machine learning and optimization tailored specifically for fog computing-based intrusion detection applications. Figure 1 shows the proposed GLIDS-Net research model.

The detailed analysis of GLIDS-Net is given as follows:

Step 1: UNSW-NB Dataset: The first step in developing an IDS for fog computing involves selecting an appropriate dataset for training and testing the intrusion detection model. One common choice is the UNSW-NB dataset, which is a well-known dataset in cybersecurity research. The UNSW-NB dataset contains a comprehensive collection of network traffic data captured from a realistic network environment, including various types of attacks and normal activities.

Step 2: Dataset Preprocessing: After selecting the UNSW-NB dataset, the next step is to preprocess the data to prepare it for feature selection and model training. Dataset preprocessing involves several tasks, including data cleaning, handling missing values, normalization, or standardization of features, and splitting the dataset into training and testing subsets. Preprocessing ensures that the data is in a suitable format for further analysis and model development.

Step 3: MABO Feature Selection: Feature selection plays a crucial role in building an effective IDS. In this step, MABO algorithm is applied for feature selection. The MABO algorithm is a metaheuristic optimization technique inspired by the behaviours of African buffalos in foraging for food. It aims to identify the most relevant features from the dataset that are informative for distinguishing between normal network traffic and malicious intrusions. MABO iteratively selects and evaluates subsets of features based on their contribution to the detection accuracy of the IDS.

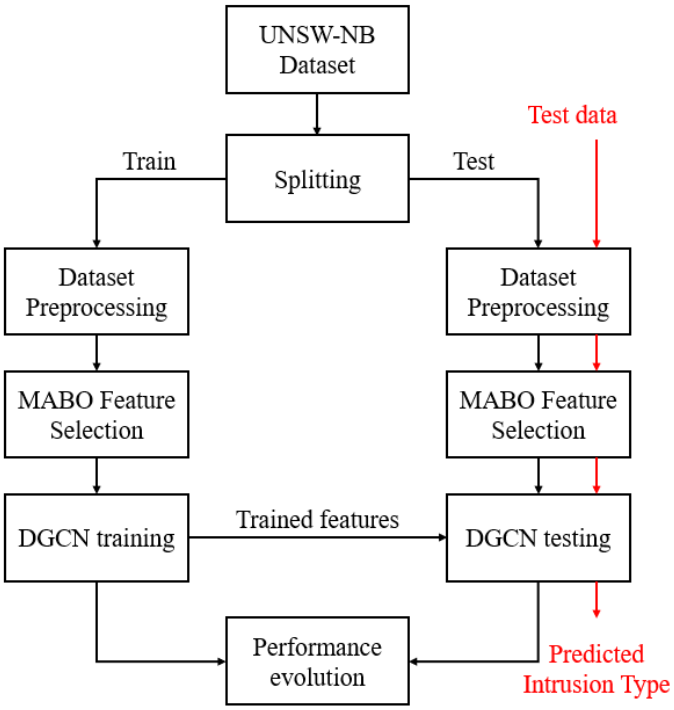


Figure 1. Proposed GLIDS-Net Architecture Design.

Step 4: DGCN Classifier: Once the relevant features are selected using MABO, the next step involves developing DGCN classifier. DGCN is a state-of-the-art deep learning architecture designed for graph-structured data, which is suitable for representing network traffic data where nodes (e.g., IP addresses, ports) and edges (e.g., connections) form a graph structure. The DGCN classifier learns hierarchical representations of network traffic patterns and correlations between features, enabling accurate intrusion detection based on learned graph embeddings.

Step 5: Prediction from Intrusion Data : In the final step, the trained DGCN classifier is used to make predictions on new or unseen network traffic data to detect potential intrusions. The prediction process involves feeding the preprocessed and feature-selected data into the trained DGCN model, which outputs predictions indicating whether each instance of network traffic is normal or indicative of an intrusion. The IDS leverages the capabilities of fog computing to perform real-time or near-real-time prediction and response, enabling rapid detection and mitigation of security threats within fog computing environments.

3.1 MABO feature selection

The MABO algorithm is a metaheuristic optimization technique inspired by the foraging behavior of African buffalos. It is designed to efficiently select the most relevant features from a dataset,

particularly in the context of machine learning tasks such as intrusion detection. The feature selection process using MABO involves several key steps and parameters, which contribute to its effectiveness in identifying informative features while minimizing computational overhead. Figure 2 shows the flowchart of MABO feature selection procedure. The detailed process illustrated as follows:

Step 1: Initialization and Parameter Setting: The MABO algorithm begins by initializing a population of candidate feature subsets. This population typically consists of potential solutions represented as binary vectors, where each element corresponds to a feature in the dataset. The algorithm also requires setting parameters such as the population size, maximum iterations, convergence criteria, and other control parameters that govern its behavior during optimization.

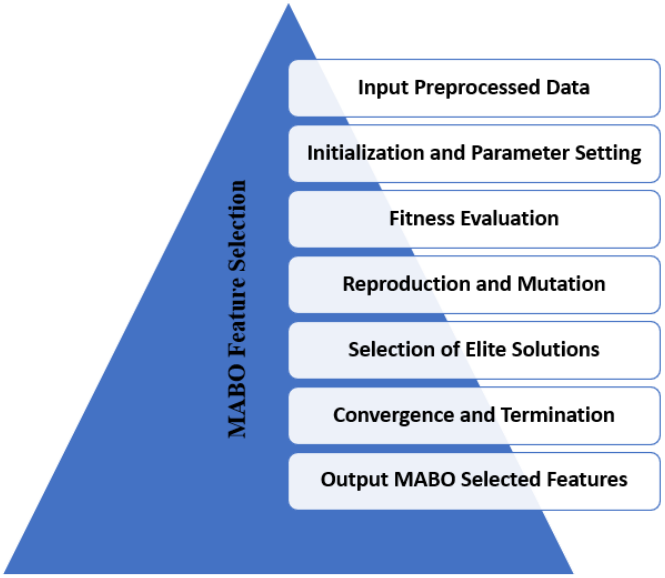


Figure 2. MABO Feature Selection Procedure.

Step 2: Fitness Evaluation: In each iteration of the MABO algorithm, the fitness of each candidate solution (feature subset) is evaluated based on a predefined fitness function. For feature selection, the fitness function measures the effectiveness of a feature subset in contributing to the performance of the machine learning model (e.g., classification accuracy, detection rate) when applied to the training dataset. The fitness evaluation process involves training the machine learning model using the selected feature subset and assessing its performance on validation data.

Step 3: Reproduction and Mutation: After evaluating the fitness of all candidate solutions, the MABO algorithm applies reproduction and mutation operators to generate new candidate solutions for the next iteration. Reproduction involves selecting promising solutions (feature subsets) based on their fitness values to create offspring solutions through crossover and mutation operations. Mutation introduces random changes to the selected feature subsets, ensuring exploration of a diverse solution space.

Step 4: Selection of Elite Solutions: During the evolution process, the MABO algorithm maintains a set of elite solutions—high-performing feature subsets that exhibit superior fitness values. These elite solutions guide the search process towards regions of the solution space that are likely to contain optimal feature subsets. Elite solutions are preserved across iterations to preserve valuable information and prevent premature convergence to suboptimal solutions.

Step 5: Convergence and Termination: The MABO algorithm continues iterating through the reproduction, mutation, and selection phases until a termination criterion is met. Common termination criteria include reaching a maximum number of iterations, achieving a desired level of fitness, or observing stagnation in the improvement of elite solutions over successive iterations. Upon convergence, the algorithm returns the best-performing feature subset (elite solution) identified during the optimization process.

3.2 DGCN classifier

The DGCNN classifier is repurposed to analyze network traffic data and identify patterns indicative of intrusions or attacks within fog computing systems. Instead of brain networks, the input data consists of network traffic graphs where nodes represent network entities (e.g., IP addresses, ports) and edges represent connections or interactions between them. Figure 3 shows the DGCNN classifier flowchart. The detailed operation of DGCNN is illustrated as follows:

Step 1: Data Representation: According to graph representation, the network traffic data is represented as a graph with nodes (network entities) and edges (connections). Each node corresponds to a specific attribute (e.g., IP address, port number) and contains features indicating the direction and nature of connections. According to node features, features associated with nodes capture information about network attributes and behaviours, such as traffic volume, protocol type, and communication patterns.

Step 2: DGCNN Model Architecture:

- **Graph Convolutional Layers**: The initial layers of the DGCNN model consist of graph convolutional layers, which operate directly on the graph structure. These layers leverage spectral graph convolution filters to process irregular graph data, allowing for convolution operations on the node features.
- **Sort Pooling Layer**: Following the graph convolutional layers, a Sort Pooling layer is employed to aggregate information from nodes and generate a fixed-size representation of the graph. This layer facilitates the integration of graph representation with traditional neural network layers.
- **1-D Convolutional Layer**: A 1-D convolutional layer is added to capture local patterns within the graph representation generated by the Sort Pooling layer. This layer extracts features from the node sequence before subsequent dense layers.
- **Dense Layers**: The network incorporates dense layers for binary categorization, aiming to classify network traffic instances as normal or intrusive. Rectified Linear Unit (ReLU) activations are used to introduce non-linearity in the convolutional and dense layers.
- **SoftMax Output Layer**: The output layer is a fully connected layer activated by SoftMax, encoding output probabilities for each class (normal or intrusive) based on learned features from the preceding layers.

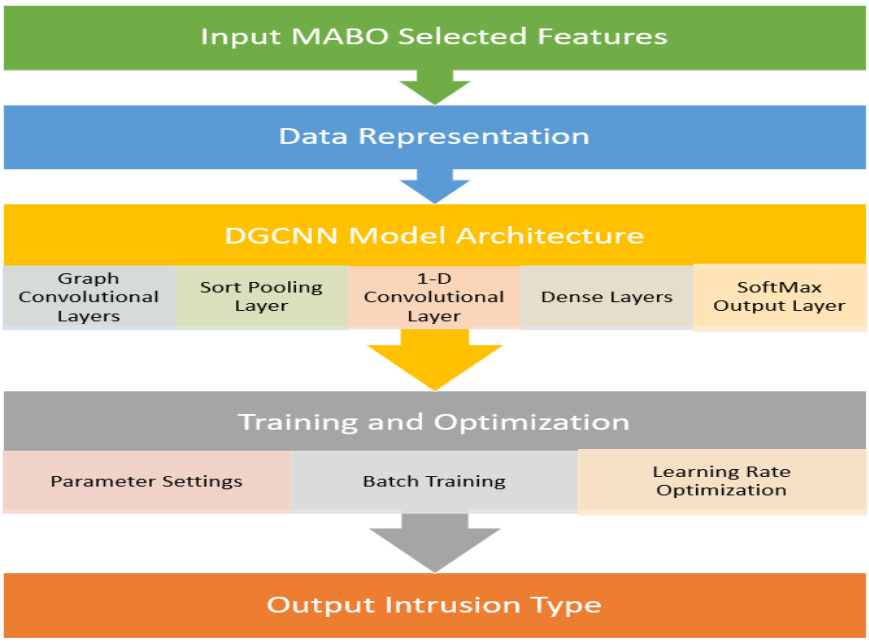


Figure 3. DGCNN Classifier Flowchart.

Step 3: Training and Optimization:

- **Parameter Settings**: Default parameters from the original DGCNN paper are utilized, including dropout regularization (dropout ratio of 0.5) between dense layers to prevent overfitting.

- **Batch Training**: Model training is conducted using batch processing (e.g., batch size of 50) over multiple epochs (e.g., 100 epochs) to optimize the network weights and biases.
- **Learning Rate Optimization**: The learning rate is a critical hyperparameter that requires optimization to ensure efficient convergence during training.

4. Results and Discussion

This section conducts a comparative analysis of different intrusion detection methods using the UNSW-NB dataset. Performance metrics such as accuracy, precision, recall, and F1-score are employed to evaluate the effectiveness of each method in detecting intrusions within fog computing environments. The results highlight the strengths and limitations of the proposed graph-based deep learning approach in comparison to traditional methods.

4.1 Performance estimation

Table 1 presents a performance comparison of the Proposed GLIDS-Net with existing approaches using the UNSW-NB dataset. The table includes key performance metrics such as accuracy, precision, recall, and F1-score, which are commonly used to evaluate the effectiveness of intrusion detection methods in fog computing environments. Starting with the SVM [19], the results show a high level of accuracy at 96.071%, along with balanced precision, recall, and F1-score metrics around 96%. The Hybrid VGG19 [13] achieves slightly higher performance with an accuracy of 96.742% and similarly balanced precision, recall, and F1-score metrics. The GAN [16] further improves the accuracy to 97.384%, demonstrating strong precision and recall rates above 98%. The RNN [17] achieves notably high accuracy at 98.676% and maintains excellent precision and F1-score metrics. The DSGRNN [20] exhibits superior performance with an accuracy of 98.949% and impressive precision, recall, and F1-score metrics above 98%.

Comparatively, the Proposed GLIDS-Net outperforms all other methods, achieving the highest accuracy of 99.01% along with exceptional precision, recall, and F1-score metrics exceeding 99%. These results highlight the effectiveness of the proposed graph-based deep learning approach in accurately detecting intrusions within fog computing environments. GLIDS-Net demonstrates superior performance in terms of both overall accuracy and the ability to identify intrusions with high precision and recall rates, showcasing its potential for enhancing cybersecurity in complex networked systems.

Table 1. Performance comparison of GLIDS-Net with existing approaches.

Method	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
SVM [19]	96.071	96.743	96.038	96.174
Hybrid VGG19 [13]	96.742	97.075	96.460	96.502
GAN [16]	97.384	98.014	96.464	96.688
RNN [17]	98.676	98.047	96.547	97.243
DSGRNN [20]	98.949	98.756	98.363	98.722
Proposed GLIDS-Net	99.01	99.10	99.13	99.12

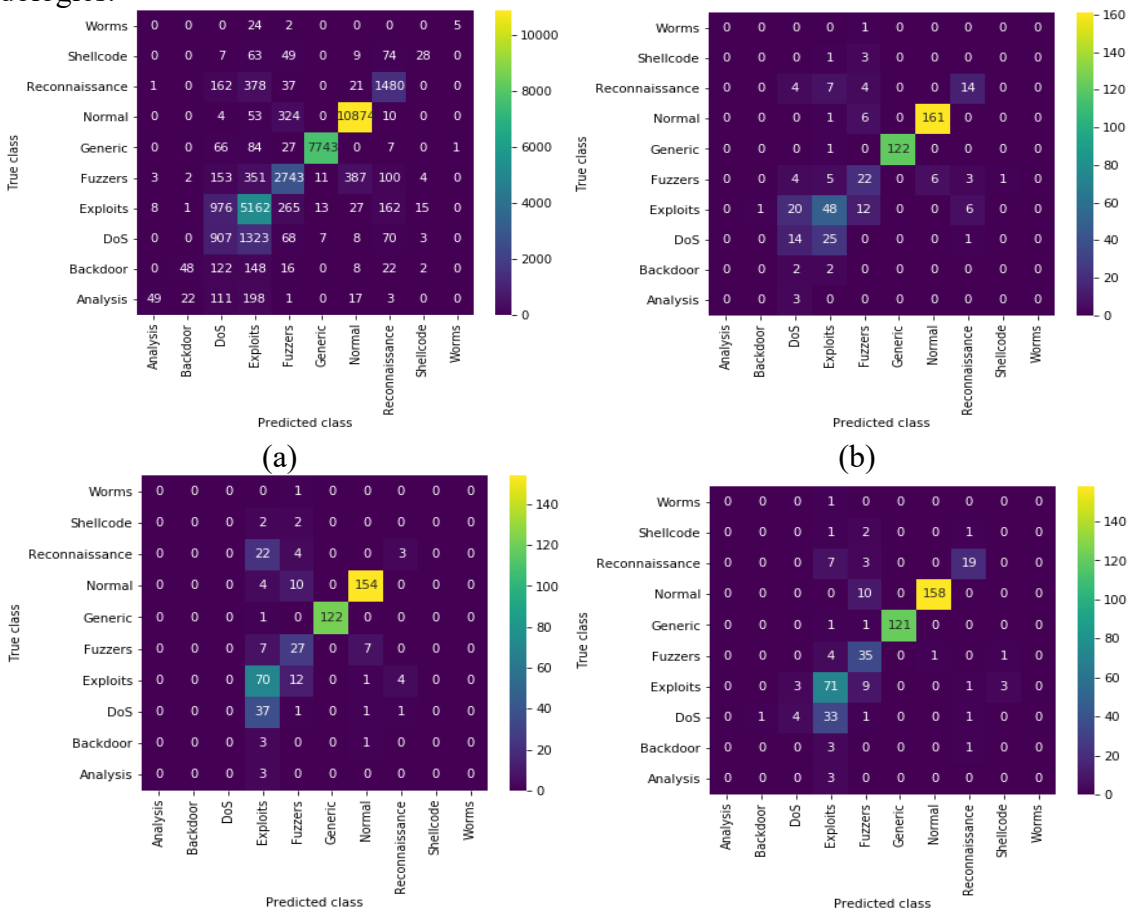
4.2 Confusion matrix and RoC curves analysis

Figure 4 displays the confusion matrices of different IDS methodologies applied to intrusion detection using the UNSW-NB dataset. Each subfigure represents the confusion matrix for a specific method, illustrating the model's performance in classifying instances of normal and intrusive network traffic. Figure 4 (a) shows the confusion matrix for the SVM approach. It indicates the true positive (TP), true negative (TN), false positive (FP), and false negative (FN) values, providing insights into the SVM's ability to accurately classify instances. The confusion matrix for the Hybrid VGG19 method demonstrated in Figure 4 (b) and the model's classification performance, highlighting its ability to distinguish between normal and intrusive network activities based on learned features. Figure 4 (c) presents the confusion matrix for the GAN approach. It illustrates how well the GAN model identifies true positives and true negatives while minimizing false positives and false negatives. In Figure 4 (d), the confusion matrix for the RNN method shows the model's classification results, indicating its

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 6, No.2, June : 2024

effectiveness in handling sequential data and detecting intrusions within network traffic. Figure 4 (e) displays the confusion matrix for the DSGRNN approach, showcasing its performance in accurately classifying instances and minimizing classification errors. Finally, the confusion matrix for the Proposed GLIDS-Net provides a comprehensive overview of the model's classification outcomes as shown in Figure 4 (f), highlighting its ability to achieve high accuracy and robustness in intrusion detection tasks.

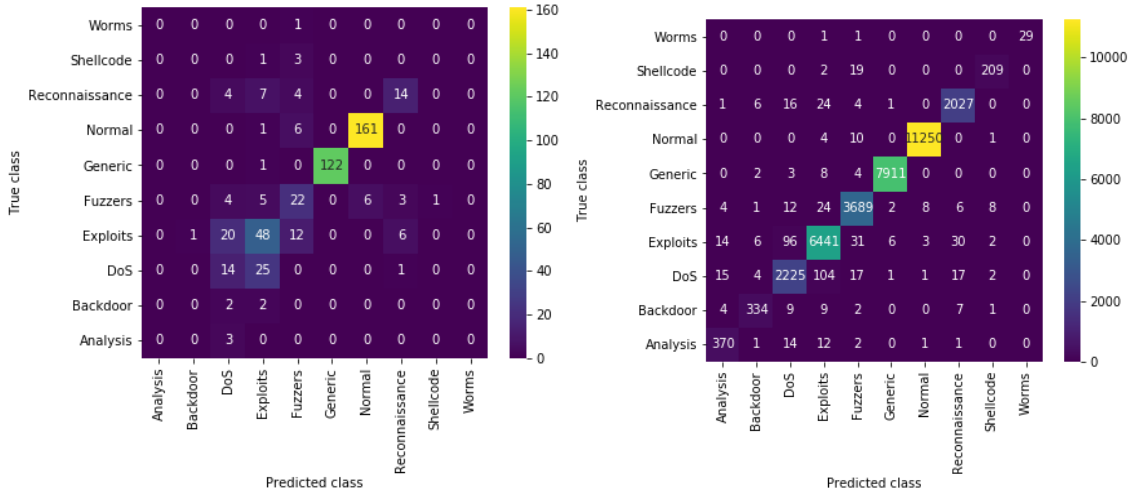
Figure 5 presents the Receiver Operating Characteristic (ROC) curves of different IDS methodologies applied to the UNSW-NB dataset for evaluating their performance in distinguishing between normal and intrusive network traffic. The ROC curve for the SVM approach is shown in Figure 5(a) illustrates how well the SVM model performs in terms of sensitivity and specificity, providing insights into its ability to make accurate classifications across different threshold levels. Figure 5(b) displays the ROC curve for the Hybrid VGG19 method, demonstrating the model's discriminatory power and trade-off between true positive and false positive rates. The ROC curve of the GAN approach (Figure 5(c)) visualizes the model's performance in distinguishing between normal and intrusive network activities, highlighting its effectiveness in minimizing false positives. Figure 5(d) shows the ROC curve for the RNN method, indicating the model's ability to balance sensitivity and specificity in intrusion detection tasks. The ROC curve for the DSGRNN approach is shown in Figure 5(e), depicts how well the model discriminates between different classes of network traffic based on learned features. Finally, figure 5(f) presents the ROC curve for the Proposed GLIDS-Net, highlighting its superior performance in accurately classifying instances of normal and intrusive network traffic compared to other IDS methodologies.



(a) (b)

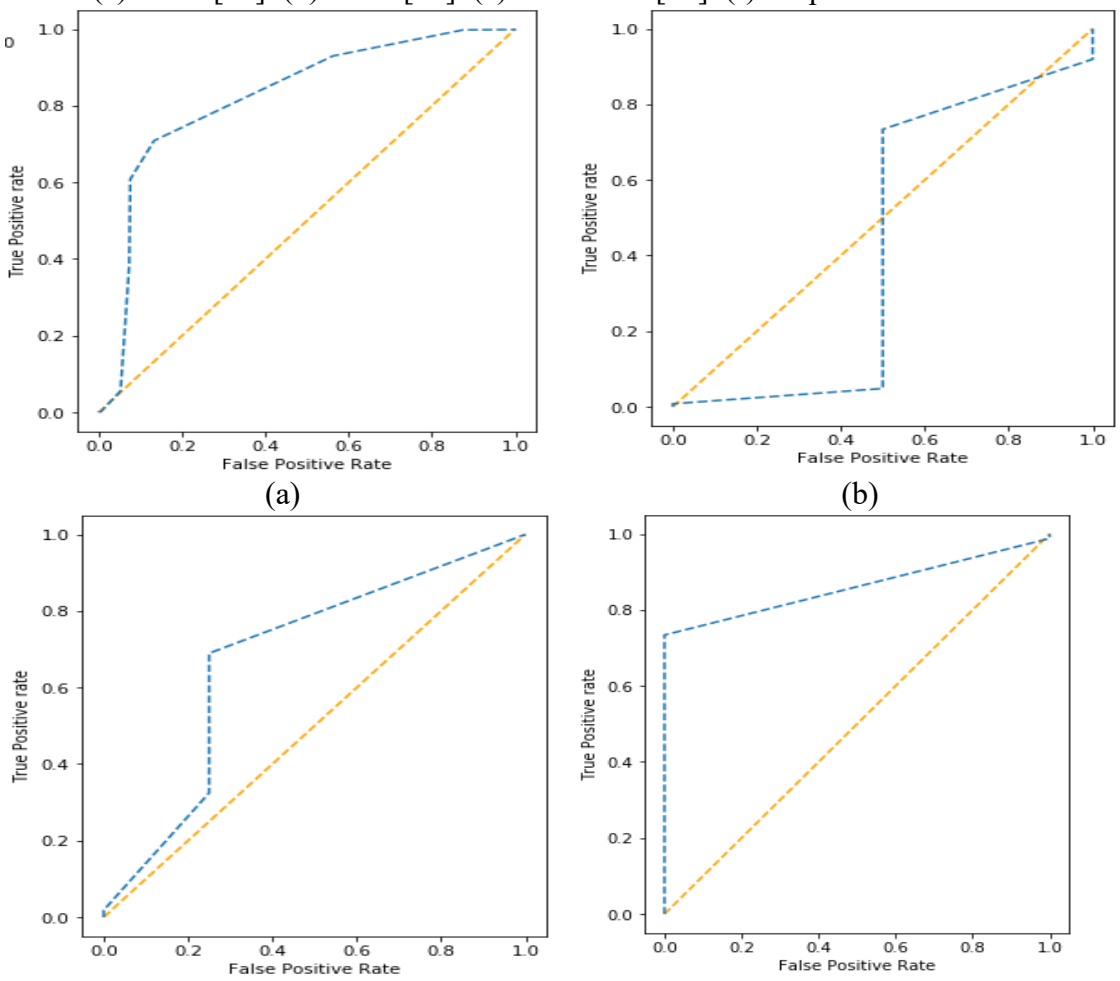
(c) (d)

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 6, No.2, June : 2024



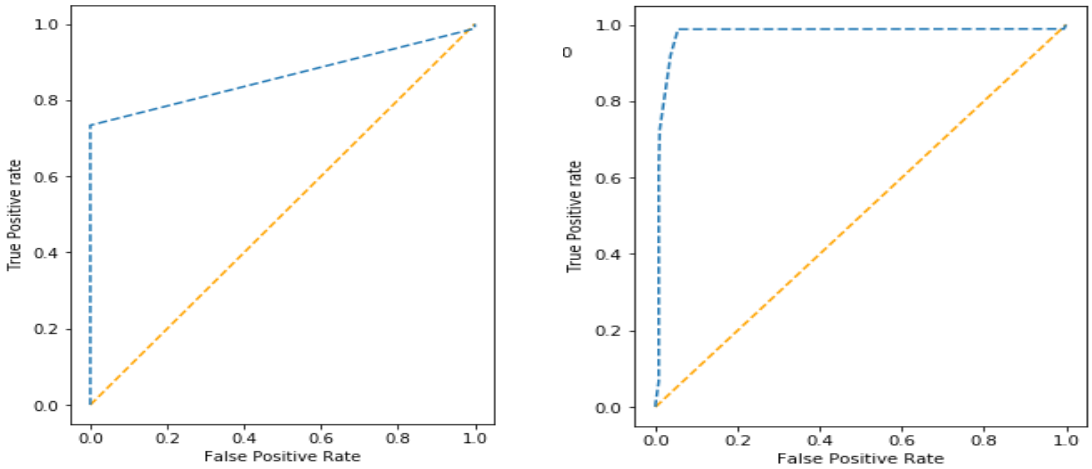
(e) (f)

Figure 4. Confusion matrices of various IDS methodologies. (a) SVM [19]. (b) Hybrid VGG19 [13]. (c) GAN [16]. (d) RNN [17]. (e) DSGRNN [20]. (f) Proposed GLIDS-Net.



(a) (b)

(c) (d)



(e) (f)

Figure 5. RoC curves of various IDS methodologies. (a) SVM [19]. (b) Hybrid VGG19 [13]. (c) GAN [16]. (d) RNN [17]. (e) DSGRNN [20]. (f) Proposed GLIDS-Net.

5. Conclusion

In conclusion, the development of an IDS tailored for fog computing environments represents a significant advancement in cybersecurity for modern networked systems. The proposed methodology, leveraging feature selection with the MABO algorithm and employing DGCN classifier, demonstrates promising capabilities for detecting and mitigating network intrusions within dynamic and distributed fog computing architectures. Looking ahead, several avenues for future research and enhancements can be explored. Firstly, further optimization and refinement of the MABO algorithm could enhance the efficiency and effectiveness of feature selection, enabling more accurate intrusion detection with reduced computational overhead. Additionally, exploring novel deep learning architectures and techniques tailored specifically for fog computing environments could lead to even more robust and scalable IDS solutions. Moreover, integrating advanced anomaly detection methods and leveraging reinforcement learning approaches for adaptive intrusion response within fog computing systems could enhance the overall security posture. Furthermore, investigating the integration of blockchain technology for ensuring data integrity and auditability in fog computing-based IDS implementations could be a promising direction.

References

[1] Jumani, Awais Khan, Jinglun Shi, Asif Ali Laghari, Zhihui Hu, Aftab ul Nabi, and Huang Qian. "Fog computing security: A review." *Security and Privacy* 6, no. 6 (2023): e313.

[2] Yi, Lizhi, Mei Yin, and Mehdi Darbandi. "A deep and systematic review of the intrusion detection systems in the fog environment." *Transactions on Emerging Telecommunications Technologies* 34.1 (2023): e4632.

[3] Sonker, Sanjay Kumar, Vibha Kaw Raina, Bharat Bhushan Sagar, and Ramesh C. Bansal. "Fog computing-based IoT-enabled system security for electrical vehicles in the smart grid." *Electrical Engineering* (2024): 1-17.

[4] Mohamed, Doaa, and Osama Ismael. "Enhancement of an IoT hybrid intrusion detection system based on fog-to-cloud computing." *Journal of Cloud Computing* 12, no. 1 (2023): 41.

[5] Tu, Shanshan, Muhammad Waqas, Akhtar Badshah, Mingxi Yin, and Ghulam Abbas. "Network intrusion detection system (NIDS) based on pseudo-siamese stacked autoencoders in fog computing." *IEEE Transactions on Services Computing* (2023).

[6] Alzahrani, Rami J., and Ahmed Alzahrani. "A novel multi algorithm approach to identify network anomalies in the IoT using Fog computing and a model to distinguish between IoT and Non-IoT devices." *Journal of Sensor and Actuator Networks* 12, no. 2 (2023): 19.

[7] Ajao, Lukman Adewale, and Simon Timothy Apeh. "Secure fog computing vulnerability in smart city using machine learning and blockchain technology." *networks* 20, no. 23 (2023).
[8] Sajid, Junaid, Kadhim Hayawi, Asad Waqar Malik, Zahid Anwar, and Zouheir Trabelsi. "A fog computing framework for intrusion detection of energy-based attacks on UAV-assisted smart farming." *Applied Sciences* 13, no. 6 (2023): 3857.
[9] Paul, P. Mano, R. Shekhar, I. Diana Jeba Jingle, and I. Berin Jeba Jingle. "Prevention and Mitigation of Intrusion Using an Efficient Ensemble Classification in Fog Computing." In *International Conference on Computer & Communication Technologies*, pp. 173-181. Singapore: Springer Nature Singapore, 2023.
[10] Zhao, Guosheng, Yang Wang, and Jian Wang. "Lightweight intrusion detection model of the internet of things with hybrid cloud-fog computing." *Security and Communication Networks* 2023 (2023).
[11] Ma, Xiaoxue, Yun Li, and Yan Gao. "Decision model of intrusion response based on markov game in fog computing environment." *Wireless Networks* 29, no. 8 (2023): 3383-3392.
[12] Ajao, Lukman Adewale, and Simon Tooswem Apeh. "Blockchain integration with machine learning for securing fog computing vulnerability in smart city sustainability." In 2023 1st International Conference on Advanced Innovations in Smart Cities (ICAISC), pp. 1-6. IEEE, 2023.
[13] Binbusayyis, Adel. "Hybrid VGG19 and 2D-CNN for intrusion detection in the FOG-cloud environment." *Expert Systems with Applications* 238 (2024): 121758.
[14] Sonawane, Sudarshan S. "Enhanced Feature Optimization for Multiclasss Intrusion Detection in IOT Fog Computing Environments." *Scalable Computing: Practice and Experience* 25, no. 2 (2024): 1246-1263.
[15] Sonker, Sanjay Kumar, Vibha Kaw Raina, Bharat Bhushan Sagar, and Ramesh C. Bansal. "Fog computing-based IoT-enabled system security for electrical vehicles in the smart grid." *Electrical Engineering* (2024): 1-17.
[16] Yao, Wei, Han Shi, and Hai Zhao. "Scalable anomaly-based intrusion detection for secure Internet of Things using generative adversarial networks in fog environment." *Journal of Network and Computer Applications* 214 (2023): 103622.
[17] Syed, Naeem Firdous, Mengmeng Ge, and Zubair Baig. "Fog-cloud based intrusion detection system using Recurrent Neural Networks and feature selection for IoT networks." *Computer Networks* 225 (2023): 109662.
[18] Chakraborty, Ananya, Mohit Kumar, and Nisha Chaurasia. "Secure framework for IoT applications using Deep Learning in fog Computing." *Journal of Information Security and Applications* 77 (2023): 103569.
[19] Azarkasb, S. O., & Khasteh, S. H. (2023). Advancing Intrusion Detection in Fog Computing: Unveiling the Power of Support Vector Machines for Robust Protection of Fog Nodes against XSS and SQL Injection Attacks. *Journal of Engineering Research and Reports*, *25*(3), 59-84.
[20] Dhiyanesh, B., A. Asha, G. Kıruthıga, and R. Radha. "Enhancıng Healthcare Data Securıty In Fog Computıng: A Deep Spectral Gated Recurrent Neural Network-Based Intrusıon Detectıon System Approach." (2024).