A Comprehensive Approach to Cyberattack Detection in Edge Computing Environments

Khder Alakkari [1], Alhumaima Ali Subhi [2], Hussein Alkattan [3], Ammar Kadi [4], Artem Malinin [4], Irina Potoroko [4], Mostafa Abotaleb [3], El-Sayed M El-kenawy [*5]

[1] Department of Statistics and Programming, Faculty of Economics, University of Tishreen, Latakia, P.O. Box 2230, Syria

[2]Department of Food and Biotechnology, South Ural State University, 454080 Chelyabinsk

[3] Electronic and Computer Center, University of Diyala, Baqubah MJJ2+R9G, Iraq

[4]Department of System Programming, South Ural State University, 454080 Chelyabinsk, Russia

[5]Department of Communications and Electronics, Delta Higher Institute of Engineering and Technology, Mansoura 35111, Egypt

Emails: khderalakkari1990@gmail.com; alhumaimaali@uodiyala.edu.iq; alkattan.hussein92@gmail.com; ammarka89@gmail.com; artemmalinin3@gmail.com; irina_potoroko@mail.ru; abotalebmostafa@bk.ru; skenawy@ieee.org

Abstract

This research is concerned with the critical domain of cybersecurity in edge computing environments, which aims to strengthen defenses against increasing cyber threats that target interconnected Internet of Things (IoT) devices. The widespread adoption of edge computing introduces vulnerabilities that necessitate a strong framework for detecting cyberattacks. This study utilizes Long Short-Term Memory (LSTM) networks to present a comprehensive approach based on stacked LSTM layers for detecting and mitigating cyber threats in the dynamic landscape of edge networks. Using the NSL-KDD dataset and rigorous experimentation, this model demonstrates its ability to detect subtle anomalies in network traffic, which can be used to accurately classify malicious activities while minimizing false alarms. The findings highlight the potential of LSTM-based approaches to enhance security at the edge, providing promising avenues for strengthening IoT ecosystems' integrity and resilience against emerging cyber threats.

Keywords: Edge Computing; Cyberattack Detection; Cyber Threats; Network Security; Edge Devices; Intrusion Detection; IoT Networks; Cybersecurity Solutions; Edge Security.

1. Introduction

The IoT landscape has seen a proliferation of interconnected devices, which has brought about a paradigm shift in computing and ushered in the era of edge computing. Edge computing is characterized by decentralized data processing at the periphery of networks, which provides unprecedented benefits in terms of latency reduction, bandwidth optimization, and real-time data analysis [1-3]. However, this decentralized architecture has also made edge devices more susceptible to cyber threats, necessitating a strong framework for cyberattack detection and mitigation within these environments [4-5]. Cybersecurity in edge computing environments is a complex problem

Doi : https://doi.org/10.54216/JCIM.130107
Received: May 28, 2023 Revised: August 16, 2023 Accepted: December 21, 2023

mainly because of the diverse and distributed nature of IoT devices [6]. The inherent limitations of edge devices such as computational power, memory, and security protocols make them highly vulnerable to various types of cyber threats including malware infiltration and distributed denial-of-service (DDoS) attacks. Addressing these vulnerabilities requires an all-inclusive approach that not only identifies and detects cyber intrusions but also integrates preemptive measures to strengthen the security posture of edge devices and networks [7].

The main objective of this paper is to explain the important aspects of cyberattack detection in edge computing environments by deeply examining the challenges, existing methodologies, and emerging strategies that can be used to strengthen security. This research aims to propose a comprehensive framework that combines state-of-the-art technologies like machine learning algorithms and anomaly detection techniques with strong security protocols designed for the specific requirements of edge devices and networks [9]. This research has practical implications for industries that rely on edge computing such as healthcare, smart cities, manufacturing, etc. By strengthening the defenses against cyber threats at the edge, it is possible to maintain data integrity and confidentiality while ensuring smooth functioning and reliability of IoT ecosystems thereby creating a more resilient and secure digital infrastructure for tomorrow.

2. The Used Methododology

The methodological approach used in this study provides a basis for a rigorous and systematic exploration of cyberattack detection in edge computing environments. The use of Long Short-Term Memory (LSTM) networks, especially the inclusion of stacked LSTM layers, is a key component of our model's architecture for detecting cyber attacks in edge computing environments [10]. LSTMs are a type of recurrent neural network (RNN) that are good at capturing temporal dependencies and long-range dependencies within sequential data, making them suitable for analyzing time-series data such as network traffic. The main idea behind LSTM is its ability to retain and selectively update information over long sequences, thereby mitigating the vanishing gradient problem encountered in traditional RNNs.

Our model's architecture exploits the novel features of stacked LSTM layers by using their hierarchical learning capabilities to capture complex patterns within the network traffic data. This process involves sequentially stacking multiple LSTM layers to enable a deeper understanding of temporal dynamics and intricate relationships within the data. This architecture helps extract complex features and representations from input data, which allows the model to discern subtle patterns indicative of cyber-attacks amidst normal network behavior [11]. The LSTM-based model is trained for attack detection using the NSL-KDD dataset, where the network traffic data is preprocessed and fed into the stacked LSTM layers. The training involves iterative forward and backward passes through the network, adjusting the model's weights to minimize the prediction errors and optimize its ability to differentiate between normal and malicious network behaviors. After training, testing data is used to rigorously evaluate the model's performance metrics, thereby validating its effectiveness in accurately identifying and classifying cyber-attacks within edge computing environments. Furthermore, hyperparameter tuning such as adjusting the number of LSTM layers, units within each layer, and optimization algorithms significantly contributes to the performance optimization of the model. The reason behind using stacked LSTM layers is that they can capture and learn complex temporal dependencies across multiple abstraction levels which enable it to discern subtle variations in network traffic indicative of cyber threats while minimizing false positives and false negatives (refer to Algorithm 1).

Algorithm 1: Procedural Design of Long Short-Term Memory Cell

1: In time t, the value of the candidate cell gets computed as follows:

$$\tilde{C}_t = \tanh\left(W_C[h_{t-1}, x_t] + b_C\right) = \tanh\left(W_{xC}x_t + W_{hC}h_{t-1} + b_C\right) \qquad (1)$$

2: The input gate i_t is managed to determine the amount of new information to insert to the cell state.

Doi : https://doi.org/10.54216/JCIM.130107
Received: May 28, 2023 Revised: August 16, 2023 Accepted: December 21, 2023

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i),\ i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \quad (2)$$

3: The forget gate f_t is employed to determine the information of the cell is to be strip off.

$$f_t = \sigma\big(W_f[h_{t-1}, x_t] + b_f\big) f_t = \sigma\big(W_{xf}x_t + W_{hf}h_{t-1} + b_f\big) \quad (3)$$

4: Imperial value C_t of the hidden layer cell is estimated as bellows:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (4)$$

where σ_t is stated as the sigmoid activation.

5: The output gate determines which parts of the cell are to be output.

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o),\ o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \quad (5)$$

6: The hidden state in LSTM is calculated as follows:

$$h_t = o_t \tanh(C_t) \quad (6)$$

7: Stack three LSTM layer with the above design From 1-6
8: Stack connected layer with 0.3 dropout ratio.
9: calculate the probability that the current samples belonging to different attacking classes.
10: Return the correct attacking class.

3. Experimental Design

This section elucidates the comprehensive approach undertaken to architect a series of controlled experiments aimed at assessing the performance, robustness, and resilience of the proposed detection framework.

3.1. Implementation Setups

The devised cyberattack detection framework's implementation setups involved a careful combination of hardware and software configurations that were designed to mimic real-world edge computing environments. Hardware provisioning was made up of a heterogeneous collection of edge devices, which included Raspberry Pi 4 Model B nodes with ARM Cortex-A72 processors, 4GB RAM, and network interfaces for connectivity. These devices were strategically placed to simulate different edge scenarios across a distributed network topology that mimicked the variability and complexity inherent in IoT ecosystems. Software configurations utilized a mix of open-source tools and custom-built applications including Ubuntu Server for edge node operating systems, Docker for containerization, and TensorFlow for machine learning-based anomaly detection algorithms. Furthermore, security protocols such as Transport Layer Security (TLS) and firewalls were put in place to strengthen the network perimeters. The orchestrated synergy between the hardware and software configurations not only facilitated the emulation of realistic edge environments but also provided a robust foundation for evaluating the performance and resilience of the cyberattack detection framework in varied operational conditions.

3.2. Evaluation Metrics

For the model evaluation process, we choose the following metrics:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (7)$$

$$\text{Recall(Sensitivity)} = \frac{TP}{TP + FN} \quad (81)$$

$$\text{Specificity} = \frac{TN}{TN + FP} \quad (9)$$

Doi : https://doi.org/10.54216/JCIM.130107
Received: May 28, 2023 Revised: August 16, 2023 Accepted: December 21, 2023

$$\text{F1} - \text{score}\ \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (10)$$

It is worth noting that all these metrics can be calculated based on the confusion matrix.

3.3. Dataset Description

The NSL-KDD dataset is a critical resource in cybersecurity research, known for its extensive coverage of network intrusion scenarios. It contains network traffic data that has been carefully classified into attack types and normal behaviors, thus covering a wide range of cyber threats with different attack complexities and strategies. It has attributes such as network protocol types, service types, and connection attributes that give a rich and multidimensional view of network interactions. Table 1 shows the summary statistics of different features of NSL-KDD data.

Table 1: Summary statistics for the NSL-KDD dataset

	count	mean	std	min	25%	50%	75%	max
duration	148517	276.77931	2.46E+03	0	0	0	0	5.77E+04
src_bytes	148517	40227.949	5.41E+06	0	0	44	278	1.38E+09
dst_bytes	148517	17088.854	3.70E+06	0	0	0	571	1.31E+09
land	148517	0.000215	1.47E-02	0	0	0	0	1.00E+00
wrong_fragment	148517	0.020523	2.40E-01	0	0	0	0	3.00E+00
urgent	148517	0.000202	1.94E-02	0	0	0	0	3.00E+00
hot	148517	0.189379	2.01E+00	0	0	0	0	1.01E+02
num_failed_logins	148517	0.004323	7.22E-02	0	0	0	0	5.00E+00
logged_in	148517	0.402789	4.90E-01	0	0	0	1	1.00E+00
num_compromised	148517	0.255062	2.22E+01	0	0	0	0	7.48E+03
root_shell	148517	0.001508	3.88E-02	0	0	0	0	1.00E+00
su_attempted	148517	0.000976	4.24E-02	0	0	0	0	2.00E+00
num_root	148517	0.273726	2.27E+01	0	0	0	0	7.47E+03
num_file_creations	148517	0.012073	5.18E-01	0	0	0	0	1.00E+02
num_shells	148517	0.000525	2.77E-02	0	0	0	0	5.00E+00
num_access_files	148517	0.004013	9.53E-02	0	0	0	0	9.00E+00
num_outbound_cmds	148517	0	0.00E+00	0	0	0	0	0.00E+00
is_host_login	148517	0.000081	8.99E-03	0	0	0	0	1.00E+00
is_guest_login	148517	0.012308	1.10E-01	0	0	0	0	1.00E+00
count	148517	83.336561	1.17E+02	0	2	13	141	5.11E+02
srv_count	148517	28.251937	7.54E+01	0	2	7	17	5.11E+02
serror_rate	148517	0.256925	4.32E-01	0	0	0	0.85	1.00E+00
srv_serror_rate	148517	0.255337	4.33E-01	0	0	0	0.91	1.00E+00
rerror_rate	148517	0.137947	3.39E-01	0	0	0	0	1.00E+00
srv_rerror_rate	148517	0.138487	3.42E-01	0	0	0	0	1.00E+00
same_srv_rate	148517	0.672983	4.37E-01	0	0.1	1	1	1.00E+00
diff_srv_rate	148517	0.067761	1.95E-01	0	0	0	0.06	1.00E+00
srv_diff_host_rate	148517	0.097441	2.59E-01	0	0	0	0	1.00E+00
dst_host_count	148517	183.92804	9.85E+01	0	87	255	255	2.55E+02
dst_host_srv_count	148517	119.46266	1.11E+02	0	11	72	255	2.55E+02
dst_host_same_srv_rate	148517	0.534521	4.48E-01	0	0.05	0.6	1	1.00E+00
dst_host_diff_srv_rate	148517	0.084103	1.94E-01	0	0	0.02	0.07	1.00E+00
dst_host_same_src_port_rate	148517	0.145932	3.09E-01	0	0	0	0.05	1.00E+00
dst_host_srv_diff_host_rate	148517	0.030584	1.09E-01	0	0	0	0.01	1.00E+00
dst_host_serror_rate	148517	0.256122	4.28E-01	0	0	0	0.6	1.00E+00
dst_host_srv_serror_rate	148517	0.251304	4.30E-01	0	0	0	0.5	1.00E+00
dst_host_rerror_rate	148517	0.13622	3.23E-01	0	0	0	0	1.00E+00

Doi : https://doi.org/10.54216/JCIM.130107
Received: May 28, 2023 Revised: August 16, 2023 Accepted: December 21, 2023

dst_host_srv_rerror_rate	148517	0.136397	3.35E-01	0	0	0	0	1.00E+00

4. Results and Discussion

The following discussion provides a detailed analysis and interpretation of these results, exploring the subtleties, implications, and limitations of the proposed framework. The learning curves of our model in Figure 1 provide a complete picture of how the training process progresses and converges. These curves represent the model's performance metrics such as accuracy, loss, and possibly other relevant evaluation metrics plotted against the number of training iterations or epochs. The convergence of these curves indicates how the model learns over time by showing how it performs on both training and validation sets. This visualization is important for assessing how well the model has performed across multiple epochs, identifying patterns of overfitting or underfitting, and finding a good trade-off between bias and variance.

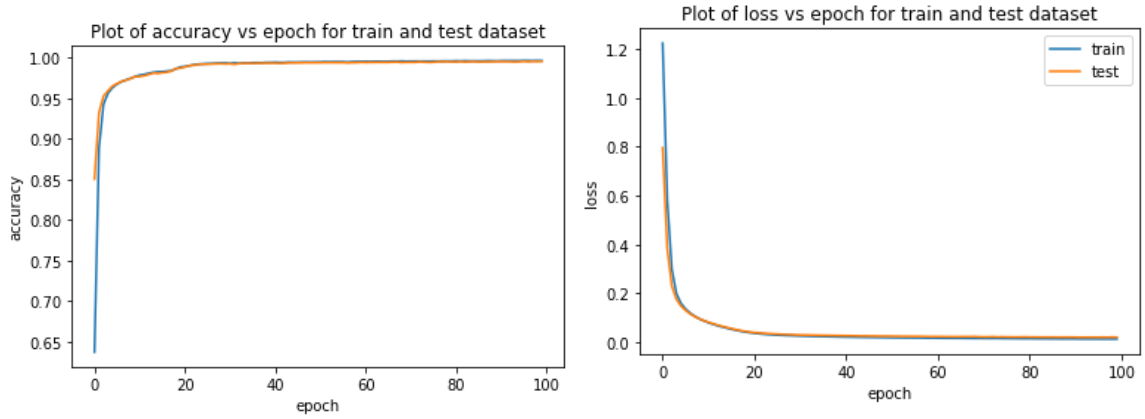

Figure 1: Evolution of Model Performance Metrics (Accuracy and Loss) Across Training Epochs

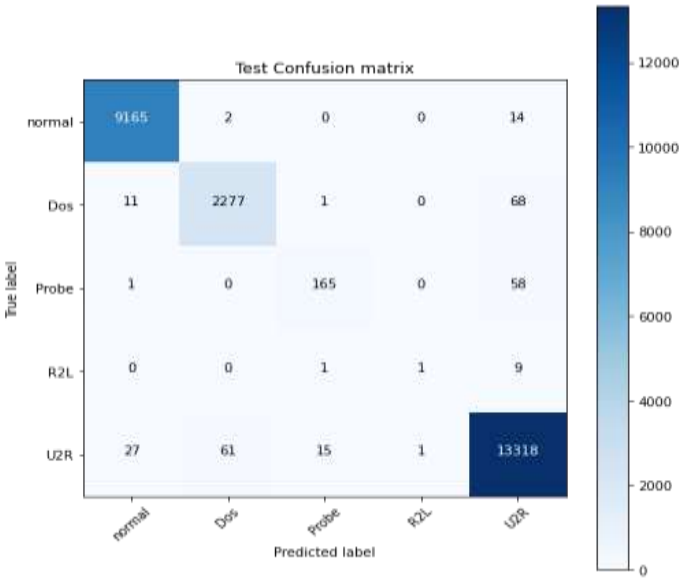

Figure 2: Confusion Matrix for Model Performance in Classifying Different

Doi : https://doi.org/10.54216/JCIM.130107
Received: May 28, 2023 Revised: August 16, 2023 Accepted: December 21, 2023

The confusion matrix in Figure 2 is derived from the model's predictions and gives a comprehensive breakdown of its classification performance across different classes. The matrix contains true positives, true negatives, false positives, and false negatives, which provide a detailed and granular evaluation of the model's predictive accuracy for each class. The confusion matrix visually represents the model's classification outcomes and allows for an in-depth analysis of its strengths and weaknesses, revealing possible areas of misclassification or bias towards certain classes. This visualization is a basic diagnostic tool that helps to assess the model's precision, recall, and overall classification efficacy, providing valuable insights for improving the model's predictive capabilities.

The Receiver Operating Characteristic (ROC) curves in Figure 3 provide a comprehensive visualization of our model's performance at different discrimination thresholds. These curves plot the true positive rate (sensitivity) against the false positive rate (1 - specificity) across different threshold values, providing a holistic assessment of the model's classification ability. The ROC curves show how well the model can distinguish between classes by illustrating the trade-off between true positive and false positive rates. Moreover, the area under the ROC curve (AUC) is a quantitative measure of how well the model discriminates, with higher AUC indicating better overall performance. This visualization helps to evaluate and compare the performance of our model with other classifiers or scenarios, giving insights into its robustness and predictive accuracy at various thresholds.

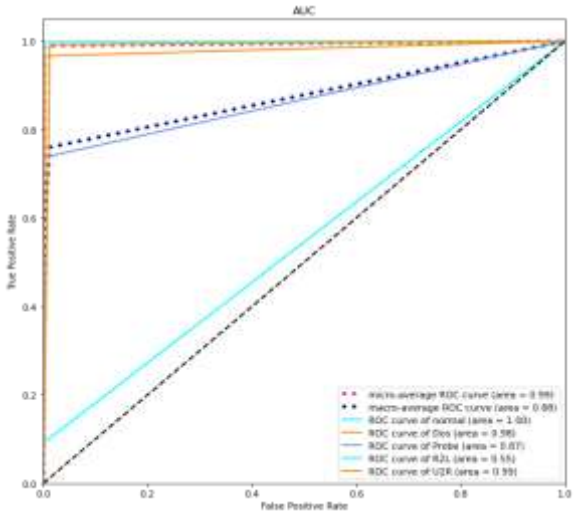

Figure 3: Receiver Operating Characteristic (ROC) Curves for illustrating Model Performance and Discrimination Thresholds Across Different Classes

5. Conclusion

This study proposes a comprehensive framework that uses stacked LSTM networks for cyberattack detection in edge computing environments. The increasing number of interconnected devices necessitates strong security measures, and our approach, which is based on LSTM's ability to understand complex temporal patterns, provides a promising way to strengthen the defenses at the edge of IoT networks. By using the NSL-KDD dataset and careful model training, validation, and evaluation, our research shows that stacked LSTM layers are effective in detecting subtle anomalies in complex network traffic. The results demonstrate that the model can accurately identify and classify cyber threats with minimal false positives and negatives, thus proving its practicality in real-world edge scenarios. However, enhancing cybersecurity in edge computing is still an ongoing process that calls for further investigation into hybrid models, ensemble techniques, and adaptive learning mechanisms to improve the robustness and scalability of intrusion detection systems.

Doi : https://doi.org/10.54216/JCIM.130107
Received: May 28, 2023 Revised: August 16, 2023 Accepted: December 21, 2023

References

[1]. Rafique, Wajid, Lianyong Qi, Ibrar Yaqoob, Muhammad Imran, Raihan Ur Rasool, and Wanchun Dou. 2020. "Complementing IoT Services through Software Defined Networking and Edge Computing: A Comprehensive Survey." IEEE Communications Surveys and Tutorials. https://doi.org/10.1109/COMST.2020.2997475.

[2]. Huong, Truong Thu, Ta Phuong Bac, Dao M. Long, Bui D. Thang, Nguyen T. Binh, Tran D. Luong, and Tran Kim Phuc. 2021. "LocKedge: Low-Complexity Cyberattack Detection in IoT Edge Computing." IEEE Access. https://doi.org/10.1109/ACCESS.2021.3058528.

[3]. Singh, Ashish, Kakali Chatterjee, and Suresh Chandra Satapathy. 2022. "An Edge Based Hybrid Intrusion Detection Framework for Mobile Edge Computing." Complex \& Intelligent Systems 8 (5): 3719–46.

[4]. Alzubi, Omar A, Jafar A Alzubi, Moutaz Alazab, Adnan Alrabea, Albara Awajan, and Issa Qiqieh. 2022. "Optimized Machine Learning-Based Intrusion Detection System for Fog and Edge Computing Environment." Electronics 11 (19): 3007.

[5]. Mohy-eddine, Mouaad, Azidine Guezzaz, Said Benkirane, and Mourade Azrour. 2023. "An Effective Intrusion Detection Approach Based on Ensemble Learning for IIoT Edge Computing." Journal of Computer Virology and Hacking Techniques 19 (4): 469–81.

[6]. Garg, Sahil, Amritpal Singh, Shalini Batra, Neeraj Kumar, and Laurence T Yang. 2018. "UAV-Empowered Edge Computing Environment for Cyber-Threat Detection in Smart Vehicles." IEEE Network 32 (3): 42–51.

[7]. Gyamfi, Eric, and Anca Jurcut. 2022. "Intrusion Detection in Internet of Things Systems: A Review on Design Approaches Leveraging Multi-Access Edge Computing, Machine Learning, and Datasets." Sensors 22 (10): 3744.

[8]. Akram, Arslan. 2022. "Comprehensive Intrusion Detection System Over Edge Computing." CAPITAL UNIVERSITY.

[9]. Tian, Zhihong, Wei Shi, Yuhang Wang, Chunsheng Zhu, Xiaojiang Du, Shen Su, Yanbin Sun, and Nadra Guizani. 2019. "Real-Time Lateral Movement Detection Based on Evidence Reasoning Network for Edge Computing Environment." IEEE Transactions on Industrial Informatics 15 (7): 4285–94.

[10]. Gopalakrishnan, T, D Ruby, Fadi Al-Turjman, Deepak Gupta, Irina V Pustokhina, Denis A Pustokhin, and K Shankar. 2020. "Deep Learning Enabled Data Offloading with Cyber Attack Detection Model in Mobile Edge Computing Systems." IEEE Access 8: 185938–49.

[11]. Kim, Ho-myung, and Kyung-ho Lee. 2022. "Iiot Malware Detection Using Edge Computing and Deep Learning for Cybersecurity in Smart Factories." Applied Sciences 12 (15): 7679.

[12]. Hilal, Anwer Mustafa, Manal Abdullah Alohali, Fahd N Al-Wesabi, Nadhem Nemri, Hasan J Alyamani, and Deepak Gupta. 2021. "Enhancing Quality of Experience in Mobile Edge Computing Using Deep Learning Based Data Offloading and Cyberattack Detection Technique." Cluster Computing, 1–12.

[13]. Li, Qianmu, Shunmei Meng, Sainan Zhang, Jun Hou, and Lianyong Qi. 2019. "Complex Attack Linkage Decision-Making in Edge Computing Networks." IEEE Access 7: 12058–72.

[14]. Abeshu, Abebe, and Naveen Chilamkurti. 2018. "Deep Learning: The Frontier for Distributed Attack Detection in Fog-to-Things Computing." IEEE Communications Magazine 56 (2): 169–75.

[15]. Ismail, M. and F.Abd El-Gawad , A. (2023) "Revisiting Zero-Trust Security for Internet of Things", Sustainable Machine Intelligence Journal, 3. doi: 10.61185/SMIJ.2023.33106.

[16]. Alotaibi, Bandar. 2023. "A Survey on Industrial Internet of Things Security: Requirements, Attacks, AI-Based Solutions, and Edge Computing Opportunities." Sensors 23 (17): 7470.

[17]. Xiao, Yinhao, Yizhen Jia, Chunchi Liu, Xiuzhen Cheng, Jiguo Yu, and Weifeng Lv. 2019. "Edge Computing Security: State of the Art and Challenges." Proceedings of the IEEE 107 (8): 1608–31.

[18]. Khan, Latif U, Ibrar Yaqoob, Nguyen H Tran, S M Ahsan Kazmi, Tri Nguyen Dang, and Choong Seon Hong. 2020. "Edge-Computing-Enabled Smart Cities: A Comprehensive Survey." IEEE Internet of Things Journal 7 (10): 10200–232.

[19]. Sharma, Rohit, and Rajeev Arya. 2021. "Secure Transmission Technique for Data in IoT Edge Computing Infrastructure." Complex \& Intelligent Systems, 1–16.

[20]. Abdel-Basset, M., Hawash, H., Moustafa, N., Razzak, I., & Abd Elfattah, M. (2022). Privacy-preserved learning from non-iid data in fog-assisted IoT: A federated learning approach. Digital Communications and Networks.

Doi : https://doi.org/10.54216/JCIM.130107
Received: May 28, 2023 Revised: August 16, 2023 Accepted: December 21, 2023