Autoencoder-Based Multi-Classification for TCP Attack Detection in Private 5G Network*

Jinha Kim[1], Seounjoon Na[1], Nakyung Lee[2] and Hwankuk Kim[1†]

[1] University of Kookmin, Seoul, South Korea
{bradypus404, rinyfeel}@kookmin.ac.kr
[2] University of Sangmyung, Cheonan, South Korea

Abstract

5G networks implemented with network slicing support diverse performance and service requirements, but the resulting complexity introduces new security threats that cannot be effectively addressed by static defense mechanisms. This paper proposes an integrated security framework for 5G slicing environments that combines slice and label-based sampling, domain-based feature grouping, lightweight Autoencoder-based feature compression, and an SVM classifier. The proposed B3 (multiAE_SVM) model achieves an F1-score of 1.0 for benign, TCP SYN, and TCP XMAS traffic, and significantly improves detection performance over single Autoencoder-based models even for challenging classes such as TCP PUSH and TCP URG. Furthermore, it demonstrates resource efficiency by maintaining a low GPU utilization of 1.1%. These results demonstrate that B3 is an optimal security solution that balances detection performance and efficiency in real-time 5G slicing environments.

Keywords: *5g Network, Private 5G Network, Network Security, Autoencoder, Classification*

1 Introduction

The 5G network is emerging as a multi-service network that supports a wide range of performance and service requirements across various industries. Network slicing technology, which divides a single physical network into multiple independent logical networks, plays a key role in realizing this 5G vision [1][2]. This technology introduces a new paradigm classed Network-as-a-Service (NaaS) [3], Which enables efficient network provisioning, efficient resource allocation, service customization, and support for diverse applications [4]. However, as services diversify and network infrastructure becomes more complex, security vulnerabilities in 5G network slicing are also increasing. Traditional network security methods, limited by static policies and single points of failure, are difficult to apply effectively in the dynamic 5G network slicing environment. In particular, network slicing introduces new security challenges at multiple levels, including inter-slice, intra-slice, and slice-specific security, which hinder the full potential of 5G networks [5].

Previous studies have mainly focused on the architecture, classification, challenges, and general security issues of 5G network slicing [5]. However, research on integrated security solutions that can maintain multi-class attack detection performance while satisfying real-time processing requirements

* Proceedings of the 9th International Conference on Mobile Internet Security (MobiSec'25), Article No. 25, December 16-18, 2025, Sapporo, Japan.

† Corresponding author

remains insufficient [6]. Autoencoder-based intrusion detection systems have shown effectiveness in anomaly detection and network attack visualization, but optimized approaches that consider the dynamic characteristics and real-time constraints of the 5G slicing environment are still lacking [7].

This paper proposes a new methodology to maintain multi-class attack detection performance while meeting real-time processing requirements in the 5G slicing environment. The proposed framework introduces a design that integrates slice- and label-based sampling, domain-based feature grouping and preprocessing, efficient feature compression using encoders of autoencoders, and SVM-based multi-class classifier training and inference. This approach preserves interactions between features and slices, thereby maintaining detection performance, improving processing speed, and optimizing memory cost.

The main contributions of this paper are as follows:

(Multi-class classification) We propose an integrated framework specialized for 5G slicing environments that simultaneously meets the requirements of multi-class attack detection and real-time processing.

(Feature compression) We present a method for extracting efficient low-dimensional latent representations from large-scale network traffic data using domain-based feature grouping and a lightweight autoencoder.

(Slice information integration) We redundantly integrate slice information into feature groups, enabling the encoder to generate representations that reflect both intra-group interactions and slice characteristics.

(Overfitting mitigation) We develop a data preprocessing strategy that reduces model dependence on specific environments and improves transferability by intentionally excluding identifiable attributes (e.g., IP addresses and ports) and focusing on behavior-based aggregate statistics.

2 Background

2.1 Security Threats in 5G Networks

5G network security faces the challenge of meeting extensive requirements for performance, cost, security, and mobility management while supporting new use cases across various industries. Since legacy networks cannot easily address these requirements, network slicing which divides a single physical infrastructure into multiple logical networks to satisfy specialized needs has emerged as a key technology [8].

ENISA's "Threat Landscape for 5G Networks Report" provides an in-depth analysis of 5G design and architecture, systematically identifying vulnerabilities, threats, and risks across the infrastructure, while also suggesting security considerations. Furthermore, European countries are reinforcing policy efforts that cover not only 5G core network components and security services but also issues such as the proliferation of IoT devices and supply chain security [9].

3GPP introduced network slicing as a core element of the 5G architecture in Release 15. By creating multiple logical networks over physical infrastructure, network slicing offers the flexibility and efficiency required to support heterogeneous service demands. However, lifecycle management of slices, as well as intra-slice and inter-slice security, introduces new threat factors. Moreover, residual vulnerabilities inherited from 4G networks may be exploited in attacks. Accordingly, 3GPP continues to evolve standards that address not only 5G network architecture and security functions but also security issues arising from IoT, Device-to-Device (D2D) communication, Vehicle-to-Everything (V2X) communication, and network slicing [10][11].

2.2 Autoencoder

An autoencoder is a neural network that compresses input data into a latent space vector and reconstructs it back to the original form, widely used for dimensionality reduction and feature extraction in unsupervised learning [12]. A typical autoencoder consists of an encoder and a decoder: the encoder extracts the main features of the input data, and the decoder reconstructs them into the original data [13]. Through this process, autoencoders effectively learn the key patterns of high-dimensional data. Instead of using a single autoencoder, this study adopted a parallel learning approach using multiple autoencoders. Network traffic data is divided into multiple feature groups, each encoded by a separate autoencoder. The outputs of these encoders are combined to form the final feature vector. This approach captures complex patterns in high-dimensional data more accurately and improves the input quality for the classifier, thereby enhancing anomaly detection performance

3 Related work

Research is actively underway to detect various security threats in 5G environments. Existing research has primarily applied machine learning and deep learning techniques to detect anomalies in network traffic.

Li et al [14]. created their own dataset using virtual simulations because publicly available datasets related to 5G networks were outdated. In this paper, they propose a two-stage intelligent model, consisting of a statistical detection model and a neural network detection model, to detect malicious DDoS attacks. Experimental results demonstrate that the proposed model can distinguish between normal and abnormal traffic and classify 21 types of DDoS attacks.

Kholidy et al [15]. use machine learning techniques to detect malicious traffic to meet the enhanced security requirements of 5G networks. They propose a hybrid mechanism that utilizes various machine learning approaches to effectively classify threats (DDoS, DoS, Normal, Reconnaissance, Theft) such as denial of service, denial of detection, and resource misuse. They integrate the Deep Extra-Trees (DET) model to ensure accuracy.

Dass et al [16]. pointed out that existing 5G anomaly detection studies have difficulty in real-time application due to low detection rates, high false positive rates, data imbalances, and high computational costs. Furthermore, the datasets used did not sufficiently reflect the characteristics of 5G. To address these issues, the research team utilized the 5G-SliciNdd dataset and introduced a hybrid feature selection method combining correlation-based filtering and binary particle swarm optimization to build an optimal feature set. They then evaluated the model using kNN, XGBoost, ANN, naive Bayes, and hard voting ensemble methods, achieving 99.8% accuracy with a low false positive rate of 0.0014

Kim et al [17]. attempted to detect attacks in a 5G network slicing environment but faced challenges such as class imbalance and handling high-dimensional traffic. The research team proposed an ensemble approach that extracts features using multiple autoencoders and applies an SVM classifier. This method achieved 89.33% accuracy on a balanced dataset, and 100% recall and 90.91% F1 score on an imbalanced dataset.

Saeid Sheikhi et al [18]. proposed an unsupervised learning-based approach to detect DDoS attacks targeting the GTP protocol in 5G core networks. The proposed model leverages the collective intelligence of multiple devices to efficiently and privately identify DDoS attacks while preserving the privacy of individual network data, effectively detecting attacks in 5G networks.

Md Sajid Khan et al [19]. pointed out that most prior works relied on statistical, machine learning, and cryptographic techniques for DDoS detection, but lacked performance impact analysis in actual 5G slicing environments and dataset construction based on slices. To overcome these limitations, their study built a 5G slice testbed, experimentally analyzed the impact of DoS/DDoS attacks on performance

indicators such as bandwidth and latency, and generated a new slice-specific dataset. They further proposed the SliceSecure model based on Bi-LSTM, achieving 99.99% accuracy.

4 Methodology

4.1 System Architecture

The objective of this study is to maintain multi-class attack detection performance in a 5G slicing environment while satisfying real-time processing requirements (latency, throughput, and resources). To this end, the proposed methodology consists of four components, as Architecture in Figure 1. (1) The dataset is constructed through slice- and label-based sampling. (2) Features and slices are grouped into semantic units based on domains, and group-specific preprocessing and normalization are performed. (3) For each group, many features are compressed using only the encoder of the autoencoder. (4) The outputs of the group-specific encoders are merged to train and infer an SVM-based multi-class classifier. This Architecture preserves the interactions between features and slices, thereby maintaining detection performance, improving processing speed, and reducing memory costs.

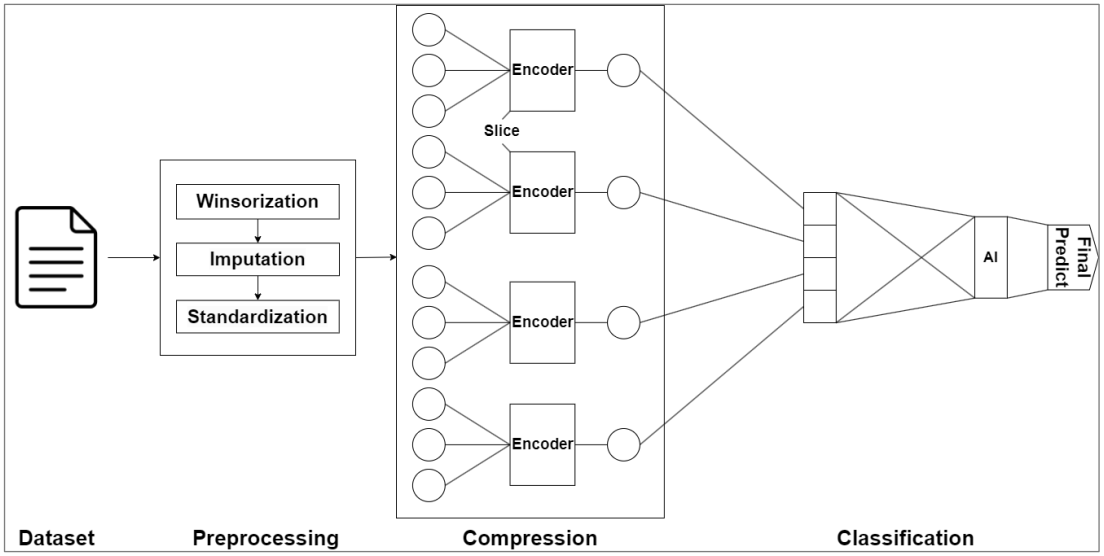

Figure 1: System Architecture

4.2 Data Preprocessing

This section describes data quantification, normalization procedures, and identifiable feature exclusion methodology.

Column names in the raw dataset are first normalized (converted to lowercase and stripped of leading and trailing spaces) to establish a consistent naming convention. Each field is then converted to the appropriate type. Fields that should be treated as numbers are converted to numeric values, and string or categorical fields are encoded as numbers as needed, except for fields with very high cardinality. Fields that are inherently numeric, such as slice values, are preserved as numeric data.

To improve the robustness of data distribution, outliers are clipped (Winsorized) based on the training set, and missing values are imputed with the mean. Group-wise standardization (scaling within groups) is then applied to align distributions inside each semantic group. This preprocessing sequence outlier removal → missing value imputation → group standardization is designed to ensure training stability of the encoder and reproducibility of latent representations.

Identifiable features (e.g., IP address, port) are deliberately excluded from the input in this study for three reasons. First, IP and port information are tightly bound to specific hosts or services, posing a high risk of overfitting to environmental characteristics. Second, such identifiers vary significantly across time and environments, reducing the transferability of learned patterns. Third, aggregated behavioral statistics (e.g., packet length, inter-arrival time, session statistics) capture the essential signals of malicious behavior more reliably. Accordingly, this study secures detection performance by leveraging aggregated and statistical features instead of identifiers.

4.3 Feature Grouping Based on Autoencoder

The rationale for adopting encoder-based compression in autoencoders can be explained from three perspectives. First, reducing the input dimensionality reduces the number of features passed to the classifier, directly reducing the computational costs of training and inference. Second, low-dimensional latent representations remove redundant and noisy information, improving classifier training efficiency and mitigating the risk of overfitting. Third, making the encoder module lightweight reduces storage requirements and runtime memory usage, thereby improving deploy ability and operability in real-time environments. For these reasons, this paper adopts a compression-based workflow that uses the encoder output as the classifier input as its core architecture.

Feature grouping using an autoencoder is based on domain semantics. Features are divided into six groups: throughput, packet size, timing, session, sub flow, and activity. Slice information, in particular, plays a crucial role in context in 5G networks, as it determines traffic processing policies, resource constraints, and priorities. Therefore, slice values are included in relevant groups, allowing the encoder to generate latent representations that simultaneously reflect intra-group interactions and slice information. Table 1 summarizes the purpose of each group and its correlation with slice information.

The architecture, which redundantly embeds slice information into volume, timing, and active groups, aims to enhance the expressive power of latent representations by allowing each group encoder to locally utilize slice context. At the same time, to mitigate the risk of overfitting due to slice inclusion, generalization is verified through group-wise ablation experiments.

Encoder-based compression first groups traffic features according to domain semantics, followed by outlier suppression, missing value imputation, and group-level standardization for each group. Then, a lightweight encoder module is applied to each group to generate low-dimensional representations. The encoders are trained jointly with the classifier in an integrated manner to align with the classification objective.

Group	Describe	Slice	Role
Volume	Quantitative data of traffic volume and load characteristics (throughput, bursts, packet rates)	Yes	Encodes slice-specific reserved bandwidth and shaping effects into load features, providing context to distinguish benign vs malicious behavior under identical traffic patterns
Timing	Statistical indicators of temporal patterns (jitter, periodicity, intermittency)	Yes	Captures slice-specific QoS scheduling differences in timing to separate timing-based attacks from legitimate high load
Active	Numerical data on session activity and idle patterns (connection duration, frequency)	Yes	Encodes slice service characteristics (short vs long sessions) to contextualize session behavior for normality assessment
Size	Statistical data on packet size distribution	No	Provides packet-level structural signals to aid detection of payload patterns and anomalous payload combinations
Session	Data on segment- and session-level transmission methods and burst behavior	No	Reflects transmission patterns to identify persistence- and transfer-method-based anomalies
Subflow	Statistical data on subflow composition and retransmission patterns	No	Leverages multi-flow structure, retransmission behavior, and subflow distribution to detect specific attacks or retransmission anomalies

Table 1: Domain based Feature Grouping

4.4 Multi-class Classification Model

The outputs of each group encoder are horizontally concatenated to form the final feature matrix. The latent vectors extracted in this way are preprocessed and scaled before being used as input for a classifier. In this study, we chose the Support Vector Machine (SVM)[20] as the primary classifier. The reasons for this choice are as follows:

- **Suitability for mixed high-dimensional and low-dimensional representations**:

  The proposed framework generates latent features from multiple groups, creating a hybrid feature space that combines high-dimensional semantic information with compact, low-dimensional embeddings. SVMs are well-suited for such data because their kernel functions effectively capture both linear and nonlinear decision boundaries, ensuring robust classification even in heterogeneous feature spaces.

- **Balance with dataset size and complexity:**

  Unlike deep learning models that require very large datasets to avoid overfitting, SVMs can achieve stable generalization with moderate-scale datasets. Given the size and complexity of traffic datasets in 5G slicing environments, SVM strikes a balance between expressive capacity and robustness, avoiding the excessive data and tuning requirements of deep neural networks.

- **Inference efficiency and simplicity:**

In real-time 5G environments, detection systems must minimize latency and resource consumption. SVMs provide relatively lightweight inference compared to deep learning approaches while still delivering strong classification accuracy [21]. This makes them suitable for deployment in scenarios with strict runtime constraints.

Accordingly, the SVM-based multi-class classifier offers a practical compromise between performance and efficiency making it a suitable choice for the proposed 5G slicing attack detection framework.

5 Experiment

5.1 Dataset

In this study, we use the dataset published by Khan et al [19]. This dataset was generated from a 5G network slice testbed consisting of Free5GC (v3.0.5) and UERANSIM (v3.1.0). Traffic was collected from two slice environments running on a total of 12 VMs (each provisioned with 2 GB of RAM and 2 CPUs). Normal traffic generated using hping3 and UDP/TCP DoS/DDoS traffic were simultaneously generated and captured as pcap files, which were converted to flows using CICFlowMaster. Of the original 84 features, we excluded features that interfered with training or overlooked attacks, leaving 54 features for model training. The final dataset consisted of 19,631 tcp_syn, 19,512 tcp_ack, 11,769 benign, 8,966 tcp_push, 8,966 tcp_urg, 8,965 tcp_fin, 8,965 tcp_xmas, and 8,964 tcp_scan flows, totaling 96,738 flows.

5.2 Test Setup

The experiments in this study consisted of three sets. First, the performance evaluation set compared the overall classification performance of the four models (B1_raw_SVM, B2_singleAE_SVM, B3_multiAE_SVM, and B4_multiSingleAE_SVM) using macro-F1 as the primary metric and supplementary metrics such as accuracy, class-specific accuracy/recall/F1, and balanced accuracy. Second, the label-specific analysis set examined detailed results for each class based on accuracy, recall, and F1 scores to identify attack-specific characteristics and differences between models. Third, the resource utilization set included memory and GPU usage to evaluate the efficiency of each model.

5.3 Evaluation Metric

The experimental environment was built on a server equipped with an AMD Ryzen 9 7950X 16-core processor (32 threads, up to 4.5 GHz) and a total of 125 GB of memory. Two NVIDIA GeForce RTX 4080 GPUs (16 GB memory each) were installed, operating under CUDA 12.2 with driver version 535.230.02. TCP flow data collected from a 5G slicing environment was used. The experiments employed key aggregated and statistical features selected by domain experts, and the *tcp_ymas* label was pre-removed depending on the experimental conditions. The dataset was constructed through stratified sampling based on slice and label, and then split into training, validation, and test sets with a ratio of 70/15/15.

- Results for Each Proposed Model

Table 1 compares the classification performance of the four models across various classes. Overall, B1 (raw SVM) achieved the highest performance, followed closely by B3 (multiAE_SVM), while B2 (singleAE_SVM) and B4 (multiSingleAE_SVM) showed lower performance. For positive, TCP SYN, and TCP XMAS traffic, B1, B2, and B3 all achieved scores above 0.99, indicating near-perfect classification, but B4 performed slightly worse for positive, with a score of 0.9792. For TCP ACK, B1 and B2 achieved nearly identical scores of 0.8372, while B3 achieved 0.8369. For TCP FIN, B1 achieved the highest score of 0.7804, while B3 achieved 0.7271, slightly outperforming B2 at 0.7169. In TCP PUSH, B3 achieved 0.9559, significantly higher than B2's 0.7016 and close to B1's 1.0000. TCP SCAN remained challenging for B1, B2, and B3, all scoring around 0.456, with B4 performing the worst at 0.0235. In TCP URG, B1 scored 0.6333, B3 0.4252, B2 0.2337, and B4 0.1914, demonstrating a clear improvement over B2. In summary, B3 maintained the second-best overall performance and showed notable improvements in more challenging classes such as TCP PUSH and TCP URG. This validates its validity as a strong candidate considering both detection performance and resource efficiency.

Label	raw_SVM	singleAE_SVM	multiAE_SVM	multiSingleAE_SVM
Benign	0.9994	0.9989	0.9989	0.9792
TCP ACK	0.8372	0.8372	0.8369	0.8137
TCP FIN	0.7804	0.7169	0.7271	0.5474
TCP PUSH	1.0000	0.7016	0.9559	0.1429
TCP SCAN	0.4564	0.4562	0.4564	0.0235
TCP SYN	1.000	0.9998	1.0000	0.9906
TCP URG	0.6333	0.2337	0.4252	0.1914
TCP XMAS	1.000	1.0000	1.000	1.000

Table 2: Results for Each Proposed Model

- Results for Labels

According to the results in Table 2, the proposed B3 model achieved perfect detection performance for Benign, TCP SYN, and TCP XMAS traffic, reaching 1.0 in Precision, Recall, and F1-score, demonstrating its ability to clearly distinguish between benign traffic and representative attacks. TCP ACK and TCP FIN also showed high Recall values of 0.9399 and 0.9242, respectively, indicating strong detection ability without missing any attacks, but the relatively low Precision suggests a slight possibility of false positives. On the other hand, TCP PUSH, TCP SCAN, and TCP URG recorded Recall values near 0.3, indicating frequent missed detections and suggesting the need for additional training data or improved feature design for these specific attack types. Overall, the B3 model achieved

Label	Precision	Recall	F1-score	Support
Benign	1.000	0.9977	0.9989	1766
TCP ACK	0.7543	0.9399	0.8369	2927
TCP FIN	0.5993	0.9242	0.7271	1345
TCP PUSH	0.9162	0.3348	0.4564	1344
TCP SCAN	0.7166	0.3348	0.4564	1344
TCP SYN	1.000	1.000	1.000	2945
TCP URG	0.7915	0.2907	0.4252	1345
TCP XMAS	1.000	1.000	1.000	1344

Table 3: Results for Labels

excellent detection performance for both benign and major attack categories, but the variation in Recall depending on the specific attack type remains an area for further improvement.

- Results for resource usage

Table 3 presents the autoencoder resource utilization of each model. Both B2 (singleAE_SVM) and B3 (multiAE_SVM) consumed approximately 4.3 GB of memory during the encoder stage, while B4 (multiSingleAE_SVM) consumed significantly less memory (approximately 2.5 GB). Among these, B3 achieved lower GPU utilization (1.1%) than B2 (1.9%), but maintained a similar GPU memory utilization (approximately 387 MB). This indicates that B3's multi-autoencoder approach achieves higher GPU utilization efficiency without increasing memory overhead.

Table 4 summarizes the test-time resource utilization. B1 (raw_SVM) consumed the most memory (4.8 GB) despite not utilizing GPU resources. B2 and B3 exhibited similar test memory requirements (4.6 GB), with B3 showing a slightly higher performance, but both models maintained negligible GPU utilization (0%). B4 again recorded the lowest test memory consumption (2.5 GB), reflecting its lightweight design. However, as shown in the classification results, its detection performance deteriorated.

Overall, the proposed B3 (multiAE_SVM) demonstrates a balanced tradeoff. Compared to B2, it reduces GPU load while maintaining competitive memory usage, and unlike B1, it avoids excessive memory requirements. This demonstrates that B3 offers a resource-efficient configuration suitable for real-time 5G slicing environments while ensuring stable detection performance.

Model	Enc1 Mem	Enc1 GPU(%)	Enc1 GPU Mem
B1_raw_SVM	NaN	NaN	NaN
B2_singleAE_SVM	4311.8	1.9	386.5
B3_multiAE_SVM	4344.1	1.1	387.0
B4_multiSingleAE_SVM	2457.7	0.0	386.0

Table 4: Autoencoder resource usage rate by model

Model	Test Mem	Test GPU(%)	Test GPU Mem
B1_raw_SVM	4833.3	0.0	386.0
B2_singleAE_SVM	4599.8	0.0	387.0
B3_multiAE_SVM	4679.2	0.0	387.0
B4_multiSingleAE_SVM	2457.7	0.0	386.0

Table 5: Test resource usage rate by model

6 Conclusion

The proposed B3 (multiAE_SVM) demonstrated balanced results in terms of performance and resource efficiency. In terms of detection performance, it maintained high accuracy for major traffic classes such as positive, ACK, SYN, and XMAS, while significantly improving over B2 (singleAE_SVM) for more demanding classes such as TCP PUSH (0.956) and TCP URG (0.425). Resource-wise, the GPU utilization was only 1.1%, approximately half that of B2, ensuring lightweight workloads while maintaining stable and consistent GPU memory usage across all models. Furthermore, the memory consumption during the evaluation phase was lower than that of B1 (raw_SVM), demonstrating superior performance-to-efficiency. These results demonstrate that the multi-autoencoder design is effective in capturing complex traffic patterns while minimizing unnecessary

resource consumption, demonstrating that the proposed B3 model is the optimal methodology for achieving both detection performance and operational efficiency in real-time 5G network slicing environments.

However, the latency aspect of the proposed model has not been extensively analyzed. In future work, we will build a simulation environment to quantitatively evaluate inference delay and scalability under real-time deployment, thereby demonstrating the model's stability and efficiency in time-sensitive 5G applications. Future research should further expand the scope and practicality of the proposed B3 (multiAE_SVM) model. First, to validate the model's generalization performance, an expanded dataset encompassing a wider range of attack types and complex attack scenarios must be developed. Furthermore, comprehensively evaluating real-time inference latency, throughput, and resource consumption in a real-world 5G testbed environment is essential to assessing deployment feasibility. Beyond the current combination of multiple autoencoder architectures and SVM classifiers, integration with federated learning, intelligent resource scheduling, and explainable AI techniques will further enhance scalability and transparency, enabling the framework to evolve into a robust security detection system applicable to distributed 5G slicing environments.

Acknowledgement

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No.RS-2024-00397469, Development of Private 5G Security Technology for Integrated Private 5G and Enterprise Network Security)

References

[1] Foukas, X., Patounas, G., Elmokashfi, A., & Marina, M. (2017). Network Slicing in 5G: Survey and Challenges. IEEE Communications Magazine. https://ieeexplore.ieee.org/document/7926923/

[2] Foukas, X., Marina, M., & Kontovasilis, K. (2017, October 4). Orion: RAN Slicing for a Flexible and Cost-Effective Multi-Service Mobile Network Architecture. *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*. https://dl.acm.org/doi/10.1145/3117811.3117831

[3] Le, L.-V., Lin, B., Tung, L.-P., & Sinh, D. (2018). SDN/NFV, Machine Learning, and Big Data Driven Network Slicing for 5G. 2018 IEEE 5G World Forum (5GWF). https://ieeexplore.ieee.org/document/8516953/

[4] Duan, Q. (2014). Network-as-a-Service in Software-Defined Networks for end-to-end QoS provisioning. *2014 23rd Wireless and Optical Communication Conference (WOCC)*. https://ieeexplore.ieee.org/document/6839919/

[5] Olimid, R. F., & Nencioni, G. (2020). 5G Network Slicing: A Security Overview. *IEEE Access*. https://ieeexplore.ieee.org/document/9099823/

[6] Alwis, C. de, Porambage, P., Dev, K., Gadekallu, T., & Liyanage, M. (2024). A Survey on Network Slicing Security: Attacks, Challenges, Solutions and Research Directions. *IEEE Communications Surveys & Tutorials*. https://ieeexplore.ieee.org/document/10242032/

[7] Alanazi, M. (2023). Machine Learning-based Secure 5G Network Slicing: A Systematic Literature Review. *International Journal of Advanced Computer Science and Applications*. https://thesai.org/Publications/ViewPaper?Volume=14&Issue=12&Code=IJACSA&SerialNo=39

[8] Messaoudi, F., Bertin, P., & Ksentini, A. (2020). Towards the quest for 5G Network Slicing. *2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC)*. https://ieeexplore.ieee.org/document/9045327/
[9] Paskauskas, R., Settembre, M., Køien, G. M., Liaqat, S., & Callegati, F. (2022). ENISA: 5G design and architecture of global mobile networks; threats, risks, vulnerabilities; cybersecurity considerations. *Open Research Europe*. https://open-research-europe.ec.europa.eu/articles/2-125/v3
[10] 3GPP TS 23.501, System Architecture for the 5G System (5GS)
[11] 3GPP TS 33.501, Security Architecture and procedures for 5G System
[12] Berahmand, K., Daneshfar, F., Salehi, E., & Li, Y. (2024). *Autoencoders and their applications in machine learning: a survey*. https://link.springer.com/article/10.1007/s10462-023-10662-6
[13] Akutsu, T., & Melkman, A. (2021). On the Size and Width of the Decoder of a Boolean Threshold Autoencoder. *IEEE Transactions on Neural Networks and Learning Systems*. https://ieeexplore.ieee.org/document/10373121/
[14] Li, M., Zhou, H., & Qin, Y. (2022). Two-Stage Intelligent Model for Detecting Malicious DDoS Behavior. *Sensors (Basel, Switzerland)*. https://www.mdpi.com/1424-8220/22/7/2532
[15] Kholidy, H., & Berrouachedi, A. (2023). *Enhancing security in 5G networks: a hybrid machine learning approach for attack classification*. https://ieeexplore.ieee.org/abstract/document/10479294/
[16] Dass, P., Rajak, A., & Tripathi, R. (2024). Machine Learning-Enabled Techniques for Anomaly Detection in 5G Networks. *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*. https://ieeexplore.ieee.org/document/10724005/
[17] Min-Gyu, K., & Kim, H. (2025). Ensemble Encoder-Based Attack Traffic Classification for Secure 5G Slicing Networks. *Computer Modeling in Engineering & Sciences*. https://file.sciopen.com/sciopen_public/1964975479982804993.pdf
[18] Sheikhi, S., & Kostakos, P. (2023). DDoS attack detection using unsupervised federated learning for 5G networks and beyond. *2023 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*. https://ieeexplore.ieee.org/document/10188245/
[19] Khan, M., Farzaneh, B., & Shahriar, N. (2022). *SliceSecure: Impact and detection of DoS/DDoS attacks on 5G network slices*. https://ieeexplore.ieee.org/abstract/document/10056693/
[20] Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*. https://link.springer.com/article/10.1007/Bf00994018
[21] Reshma, R., & Narawade, V. (2024). Enhancing Image Classification Accuracy With A Lightweight Hybrid Densenet And Machine Learning Model. *Nanotechnology Perceptions*. https://nano-ntp.com/index.php/nano/article/view/2115