Toward Agentic AI in 6G Security: Specialized Small Language Model for False Base Station Detection *

I Wayan Adi Juliawan Pawana[1,2], Vincent Abella[2], Hoseok Kwon[2], Bonam Kim[2], and Ilsun You[2†]

[1] Department of Electrical Engineering, Udayana University, Badung, Indonesia
adijuliawanpawana@unud.ac.id
[2] Department of Cyber Security, Kookmin University, Seoul, South Korea
{adijuliawan, vincent, hoseok1997, kimbona, isyou}@kookmin.ac.kr

Abstract

As 6G networks promise ultra-dense connectivity and real-time communication, they also face heightened security challenges, particularly the threat of False Base Stations (FBS). FBS attacks exploit protocol vulnerabilities to impersonate legitimate network nodes, enabling eavesdropping, phishing, and denial-of-service attacks. Traditional detection approaches often rely on structured numerical features and centralized infrastructure, which can limit responsiveness and scalability. In this work, we introduce a novel FBS detection framework using a Specialized Small Language Model (SLM) fine-tuned on both Radio Resource Control (RRC) and Non-access stratum (NAS) signaling data. By transforming protocol-layer features into textual representations, our approach enables a single lightweight model to learn cross-layer signaling patterns indicative of malicious activity. Experiments using the Gemma3-270M-Instruct model, fine-tuned with LoRA, demonstrate strong performance, achieving up to 85.45% validation accuracy and 86.5% F1 score on the test data. The unified architecture allows the model to capture inter-layer inconsistencies that may indicate sophisticated FBS behavior, while maintaining efficiency suitable for on-device deployment. This work highlights the potential of compact, domain-adapted language models as a foundation for agentic AI in securing future 6G networks.

Keywords: 6G Networks, Agentic AI, False Base Station, Small Language Model (SLM)

1 Introduction

The rapid evolution of wireless communication has reached a major milestone with the deployment of fifth-generation (5G) networks, which are transforming global connectivity by delivering enhanced mobile broadband, ultra-reliable low-latency communication, and massive machine-type communication. These technological advances enable an unprecedented number of devices to connect seamlessly, powering applications across industries and everyday life. However, the increasing number of connected devices inevitably broadens the attack surface, thereby creating additional opportunities for malicious exploitation. One critical and growing threat in this context is the False Base Station (FBS), a rogue network node that deceives user equipment (UE) into connecting with it. Once successful, FBS can conduct attacks such as eavesdropping, phishing, man-in-the-middle manipulation, and denial-of-service, putting user privacy and network security at risk. Recent real-world cases, including large-scale phishing campaigns in Thailand [1, 2] and the seizure of SMS "blaster" devices in multiple countries [3], demonstrate

*Proceedings of the 9th International Conference on Mobile Internet Security (MobiSec'25), Article No. 44, December 16-18, 2025, Sapporo, Japan.

†Corresponding author

the severity and global reach of this issue. As networks become more densely connected and dynamic, developing effective strategies to detect and counter FBS threats is vital for ensuring the resilience and trustworthiness of future 6G systems.

Detecting FBS requires identifying features that distinguish them from legitimate network nodes. Such indicators include anomalous signaling patterns, irregular transmission power levels, atypical network identifiers, and inconsistencies in temporal behavior. Based on these characteristics, several detection mechanisms have been proposed, which can broadly be categorized into network-side, radio access network (RAN)-side, and user equipment (UE)-side approaches. Network-based solutions leverage centralized monitoring and coordination but often face scalability and latency constraints. In contrast, UE-side detection enables decentralized, real-time identification of suspicious activity through local observations, thereby improving responsiveness and strengthening resilience against dynamic and localized threats.

In recent years, machine learning (ML) has gained traction as a promising approach for FBS detection due to its ability to learn complex patterns and adapt to evolving attack strategies. By analyzing diverse features such as transmission anomalies, signaling irregularities, and temporal deviations, ML models have demonstrated strong performance in distinguishing legitimate base stations from malicious ones. However, most existing ML approaches are designed around structured numerical data and predefined features, which may limit their flexibility in capturing the broader contextual patterns associated with increasingly sophisticated FBS attacks.

To address this gap, we propose the use of Small Language Models (SLMs) for FBS detection. SLMs are compact language models designed to operate efficiently on consumer-grade or resource-constrained devices, delivering low-latency inference suitable for real-time applications. Their ability to capture sequential and contextual patterns makes them well suited for analyzing signaling irregularities, message content, and communication sequences associated with FBS activity. Furthermore, their lightweight architecture enables deployment directly on user equipment or distributed network nodes, thereby reducing reliance on centralized infrastructure while supporting scalable and responsive detection mechanisms.

The remainder of this paper is organized as follows: Section II provides an overview of the background and related work on false base station detection. Section III details the proposed methodology, while Section IV presents the experimental result. Finally, Section V concludes the paper.

2 Background and Related Works

The detection of FBS is critical for ensuring the security of mobile networks, particularly in the context of emerging 5G and future 6G technologies. As a result, FBS detection has become a prominent area of research in mobile network security. Various methodologies have been proposed to address this issue, including network-based detection, specification-based approaches, machine learning techniques, and cryptographic solutions.

Nakarmi et al. [4] introduced a network-based detection system for 3GPP technologies that operate without requiring user-installed software, leveraging network topology and configuration information for more reliable detection. Their system was successfully implemented and validated in both laboratory settings and real-world trials, leading to its adoption by the 3GPP standardization organization. FBSleuth [5] presents a forensic framework that utilizes unique radio frequency fingerprints to accurately identify and associate FBS devices with criminal activities. Bolcek et al. [6] use In-phase and Quadrature (IQ) data of a radio frequency (RF) signal to identify a device working as FBS. This method enhances security by effectively detecting FBSs in flexible, multi-vendor network setups.

SMDFbs [7] is a behavior rule specification-based FBS detection system that derives behavior rules from the normal operations of base stations and converts these rules into a state machine. Based on this state machine, the system detects network anomalies and mitigates threats. Addressing the root vulnerability of FBSs, a hierarchical identity-based signature protocol called Schnorr-HIBS [8] was proposed to provide efficient broadcast authentication with minimal overhead. The protocol outperforms existing 3GPP authentication methods by achieving over six times faster cryptographic delays and reducing communication costs by 31%.

Machine learning techniques have been applied to detect FBSs [9, 10]. By utilizing robust features based on Reference Signals Received Power (RSRP) and various ML algorithms, such as Regression Clustering, Anomaly Detection Forest, Autoencoder, RCGAN, and XGBoost, several approaches have shown high precision in detection even when the false base station is using a legitimate PCI. Furthermore, Mubasshir et al. [11] propose FBSDetector, an effective and efficient detection solution that can reliably identify FBSes. Unlike many network-side solutions, FBSDetector operates on the UE side, leveraging layer-3 network traces and applying ML models to uncover malicious behaviors.

3 Methodology

This study presents an approach for detecting false base stations (FBSs) using a small language model. The overall architecture of the proposed method is illustrated in Fig. 1. This section provides a detailed overview of the workflow, including the dataset, textual representation of numerical features, the small language model, and its fine-tuning process of the small language model.

3.1 Dataset

The dataset used in this study was obtained from prior research conducted on the POWDER testbed, which allows for controlled emulation of realistic cellular network environments. To ensure clarity, the dataset preparation process follows three main stages: (i) dataset generation, (ii) preprocessing, and (iii) labeling.

3.1.1 Dataset Generation

The dataset, originally introduced in [11], was generated by deploying cellular networks in the POWDER testbed. This setup included legitimate base stations (BSs), false base stations (FBSs), and mobile subscriber adversaries, with packets from all network components captured to provide a comprehensive record of signaling exchanges.

A key aspect of dataset generation is the modeling of attacker abilities, which were defined across five sophistication levels:

- Level 0: Naïve attackers who deploy FBS with excessively high transmission power.
- Level 1: Attackers who configure FBS with optimal power levels to trigger handovers without detection.
- Level 2: Attackers capable of fully cloning legitimate BS parameters, including identifiers (Cell ID, MCC, MNC, TAC, PCI), frequency and bandwidth settings, radio characteristics (e.g., transmission power, synchronization signals), and network information such as PLMN and neighbor cell details.

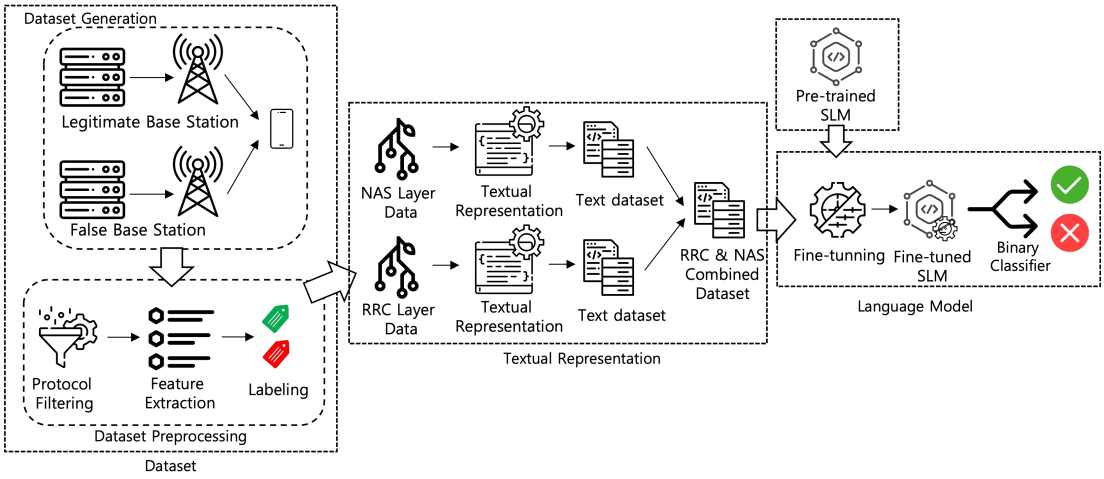


Figure 1: Methodology of this research

- Level 3: Attackers who combine Level 2 cloning with multi-step attacks (MSAs) that exploit FBS for advanced malicious actions.
- Level 4: Adaptive adversaries who evade conventional defenses by modifying non-critical fields in malicious messages (e.g., reject causes, optional attach fields) or reshaping temporal sequences of signaling messages to create deceptive patterns.

This layered attacker model ensures that the dataset covers both basic and highly sophisticated FBS threats, making it suitable for evaluating advanced detection methods. In addition, mobility scenarios were incorporated using POWDER's mobile endpoints, allowing UEs to move between cells and undergo handovers. This prevents benign mobility-induced handovers from being misclassified as malicious activity.

3.1.2 Dataset Preprocessing

To prepare the dataset for machine learning, raw packet traces were filtered to retain only signaling traffic relevant to FBS detection. Specifically, Radio Resource Control (RRC) and Non-Access Stratum (NAS) layer packets were isolated, as these layers contain critical control-plane information that can reveal malicious behavior. From these filtered packets, features were extracted: 119 fields from NAS packets and 183 fields from RRC packets, forming a structured feature set for training detection models.

3.1.3 Labeling

Finally, the dataset was labeled to support supervised learning tasks. Each sample was annotated according to its source and purpose of generation, distinguishing legitimate communication from FBS-induced or MSA-related activity. Formally, the dataset is represented as $\text{FBSAD} := \langle X_{\text{FBSAD}}, Y_{\text{FBSAD}} \rangle$, where X_{FBSAD} represents the extracted packet-level features and Y_{FBSAD} denotes the corresponding labels. Specifically, $Y_{\text{FBSAD}} = 0$ if X_{FBSAD} corresponds to a benign packet, and $Y_{\text{FBSAD}} = 1$ if the packet was generated by a FBS. This structured

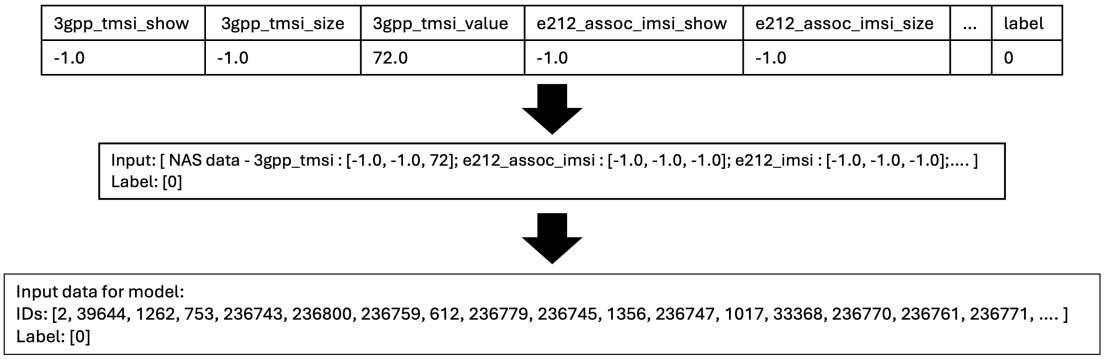


Figure 2: Dataset Transformation

labeling enables the dataset to be directly applied for training and evaluating machine learning and small language model (SLM)-based detection methods.

3.2 Textual Representation

A central aspect of the system design involves converting unstructured numerical network data, specifically derived from NAS (Non-Access Stratum) signaling messages, into a format that can be effectively interpreted by a small language model. Although language models are highly proficient at understanding and reasoning over natural language, they are not inherently optimized to capture the semantic relationships between raw numerical features. This presents a challenge in scenarios where the dataset is predominantly numeric, limiting the direct applicability of standard NLP techniques.

To address this, the NAS dataset is preprocessed by transforming numerical values into a textual format. Each value is contextualized by appending its corresponding feature name, and the resulting name-value pairs are concatenated to form a structured textual representation. This transformation enhances the model's ability to interpret feature relationships and contextual information. An illustrative example of this transformation process is shown in Fig. 2.

3.3 Small Language Model

Large Language Models (LLMs) such as GPT and LLaMA have demonstrated remarkable capabilities in natural language processing, reasoning, and multi-modal applications. Their versatility and fluency have driven rapid adoption across domains, from conversational systems to code generation. However, these advances come at the cost of enormous computational and memory requirements, often demanding specialized hardware and high energy consumption. Such constraints limit their applicability in real-time, resource-constrained environments, particularly when deployed on UE or edge devices in wireless networks. This mismatch between the scale of LLMs and the practical needs of many agentic applications motivates the exploration of lighter alternatives.

Small Language Models (SLMs) represent one such alternative. By definition, an SLM is a language model compact enough to run on consumer-grade hardware while maintaining inference latency low enough to be practical for serving the requests of a single user [12]. Unlike general-purpose LLMs, SLMs are optimized for efficiency and targeted performance rather than maximal generality. They are sufficiently powerful to handle the constrained, domain-specific

tasks often required in agentic systems while being inherently more operationally suitable for deployment in distributed and resource-limited environments. Their compactness also ensures lower energy usage and reduced deployment costs, making them a pragmatic choice for applications where scalability and responsiveness are critical.

In this context, SLMs offer several advantages over their larger counterparts. First, they provide adequate task performance for specialized domains without the overhead of maintaining broad generality. Second, their operational efficiency enables low-latency inference on edge devices, making them well aligned with decentralized agentic architectures. Third, they are cost-effective, reducing both computational and financial overhead in comparison to large-scale LLM deployments. Collectively, these characteristics suggest that SLMs are not only sufficient but often preferable for agentic AI applications. We therefore contend that SLMs will play a central role in the future of agentic AI, particularly in domains such as wireless communication security, where lightweight, autonomous, and context-aware agents are required to operate at scale.

Concrete examples highlight the promise of this direction. Google's Gemma 3 270M [13], with only 270 million parameters, demonstrates strong instruction-following and text structuring abilities while operating with extreme energy efficiency; on-device tests show negligible battery usage, making it highly practical for mobile and edge deployment. Similarly, Qwen3-0.6B [14], a 0.6 billion parameter model, combines compactness with advanced reasoning capabilities, dual "thinking" and "non-thinking" modes for adaptive performance, long-context support, and multilingual coverage. Both models illustrate how modern SLMs achieve a balance between efficiency and capability, validating their role as foundational tools for future agentic AI applications.

3.4 Fine-Tuning with LoRA

To maximize the effectiveness of FBS detection, SLMs must be adapted to the domain-specific characteristics of cellular signaling data. This adaptation is achieved through fine-tuning on curated datasets that capture representative patterns of both legitimate and malicious base station behavior.

However, deploying SLMs directly on UE comes with significant computational constraints. This makes parameter-efficient fine-tuning (PEFT) techniques especially appealing, as they adapt models to new tasks while keeping resource usage manageable. Among these methods, Low-Rank Adaptation (LoRA) has gained wide adoption due to its balance between efficiency and adaptability.

Instead of updating all parameters of a weight matrix during fine-tuning, LoRA introduces a low-rank update $\Delta W = AB$ added to the frozen pretrained weights W. With $A \in R^{d\times r}$ and $B \in R^{r\times k}$, the number of trainable parameters is reduced from $O(dk)$ to $O(r(d + k))$, where $r \ll \min(d, k)$. This greatly lowers memory footprint and training cost while preserving the ability to capture task-specific patterns. Importantly, LoRA modules can be merged back into the original weights at inference, so no extra latency is introduced.

Two hyperparameters govern its effectiveness: Rank (r) and Alpha(α). Rank controls the expressive capacity of the updates, smaller values enable lightweight adaptation, while larger values capture more complex patterns at higher cost. Alpha scales the updates, with lower values preserving generalization and higher values driving stronger task-specific adaptation. Together, these hyperparameters define the trade-off between efficiency and adaptability in LoRA-based fine-tuning.

4 Experiment

4.1 Dataset

The dataset was subsequently labeled and divided into three subsets, following an 80%–10%–10% split for training, evaluation, and testing. As outlined in Table 1, the training set contains a mix of Normal and Anomaly samples, the evaluation set provides a balanced portion for model tuning, and the testing set includes the remaining samples for final performance assessment.

Table 1: Dataset Information

Label	**Training**	**Evaluation**	**Testing**
Normal	1269	151	163
Anomaly	872	117	105
Total	2141	268	268

4.2 Implementation Details

In this study, we fine-tune a single small language model, the Gemma 3 270M Instruct, using LoRA (Low-Rank Adaptation) to enhance computational efficiency while preserving model adaptability. The fine-tuning process is conducted with a learning rate of 0.0002, for 10 training epochs, using a weight decay of 0.01 for regularization, and leveraging the paged AdamW 32-bit optimizer.

To analyze the impact of different LoRA configurations, we perform additional fine-tuning experiments using LoRA ranks of 8, 16, 32, 64, and 128, with the scaling factor (alpha) set to one-fourth of the respective rank (i.e., 2, 4, 8, 16, and 32, respectively). This experiment aims to evaluate the trade-offs between model performance, parameter efficiency, and computational cost, providing insights into the optimal LoRA configuration for small model adaptation.

The fine-tuning experiments were conducted out on a computer with an Intel Core i7-13700K CPU (3.40 GHz), 64 GB of RAM, and an NVIDIA GeForce RTX 4090 GPU (24 GB), running on a 64-bit Ubuntu 24.01.1 LTS operating system. The fine-tuning was implemented using PyTorch version 2.8.0 as the deep learning framework.

4.3 Evaluation Metrics

To evaluate the performance of classification models, we adopt a range of metrics derived from the confusion matrix. This matrix provides a summary of prediction outcomes by comparing actual class labels with those predicted by the model, as illustrated in Table 2. In the context of binary classification, we categorize the outcomes into four primary types: True Positives (TP), where the model correctly classifies positive instances; False Positives (FP), where the model incorrectly predicts negative instances as positive (Type I error); True Negatives (TN), where the model correctly identifies negative instances; and False Negatives (FN), where the model mistakenly classifies positive instances as negative (Type II error).

Accuracy (ACC) is the proportion of correct predictions among all predictions, providing an overall sense of model performance. Precision (PR) focuses on the proportion of predicted positive cases that are actually positive, indicating the accuracy of positive predictions. Recall

Table 2: Confusion Matrix

	Predicted as Positive	Predicted as Negative
Labeled as Positive	TP	FN
Labeled as Negative	FP	TN

(RC) measures the model's ability to correctly identify actual positive cases, indicating how well it captures all positives. False Positive Rate (FPR), on the other hand, indicates the proportion of actual negatives that were incorrectly classified as positive, reflecting the model's tendency to generate false alarms.

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$RC = \frac{TP}{TP + FN} \tag{2}$$

$$FPR = \frac{FP}{FP + TN} \tag{3}$$

$$PR = \frac{TP}{TP + FP} \tag{4}$$

$$F1\ Score = \frac{2 * PR * RC}{PR + RC} \tag{5}$$

5 Result and Discussion

5.1 Experimental Result

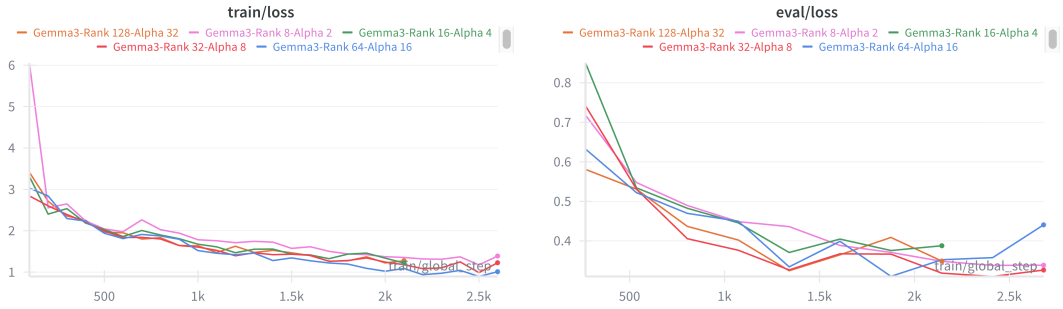
To assess the effectiveness of our approach, we fine-tuned the Gemma3-270M-Instruct using LoRA on signaling data composed of both RRC and NAS messages. The dataset captures fine-grained packet-level interactions between mobile devices and base stations, enabling the model to learn subtle differences between legitimate and malicious communication. By transforming structured numerical features from both RRC and NAS layers into textual representations, the model could process and reason over signaling patterns using natural language modeling techniques.

Table 3 summarizes the training results for five configurations with varying LoRA rank and alpha values. The configuration with rank 32 and alpha 8 achieved the highest validation accuracy at 85.45 percent, along with an F1 score of 84.2105 percent, precision of 80 percent, and recall of 88.8889 percent. This model also recorded the lowest validation loss of 0.326, suggesting that it achieved the best balance between learning signal patterns and avoiding overfitting. The model with rank 64 and alpha 16 also performed strongly, with slightly lower accuracy but a higher recall of 0.906, indicating improved sensitivity to FBS activity.

Figures 3a and 3b illustrate the training and evaluation loss curves, showing stable convergence across all configurations. The models generally reached convergence within 8 to 10 epochs, and the small gap between training and validation losses reflects good generalization. Figure 3, which shows accuracy progression over epochs, confirms that both rank 32 and rank

Table 3: Training Result

Rank	Alpha	Final Epoch	Train Loss	Val Loss	Accuracy	F1	Precision	Recall
8	2	10	1.3885	0.338541	82.4627%	80.9717%	76.9231%	85.4701%
16	4	8	1.2194	0.387776	83.9552%	83.3977%	76.0563%	92.3077%
32	8	10	1.2264	0.326325	85.4478%	84.2105%	80%	88.8889%
64	16	10	1.0092	0.440678	85.0746%	84.127%	78.5185%	90.5983%
128	32	8	1.2745	0.348994	83.9552%	82.7309%	78.0303%	88.0342%



(a) Training Loss (b) Evaluation Loss

Figure 3: Graph Training & Evaluation Loss

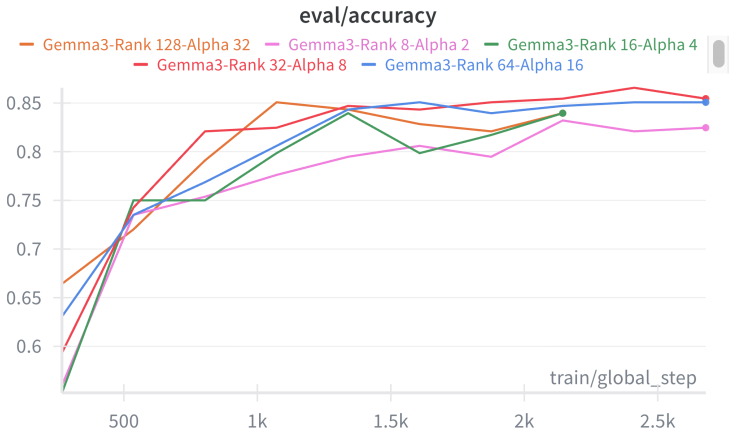
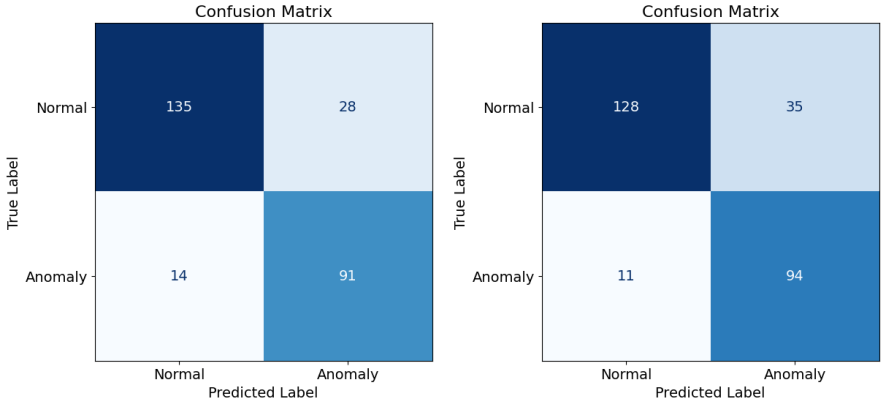


Figure 4: Accuracy Graph

64 configurations consistently improved across training iterations. This trend suggests that the model successfully learned useful patterns from both RRC and NAS signaling traffic.

Table 4: Testing Result

Rank	Alpha	Accuracy	Precision	Recall	False Positive Rate	F1 Score
8	2	80.9701%	90%	77.3006%	13.3333%	83.1683%
16	4	82.4627%	90.8451%	79.1411%	12.3810%	84.5902%
32	8	82.8358%	92.0863%	78.5276%	10.4762%	84.7682%
64	16	84.3284%	90.6040%	82.8221%	13.3333%	86.5385%
128	32	81.3433%	87.9195%	80.3681%	17.1429%	83.9744%



(a) Rank 64 and Alpha 16 (b) Rank 32 and Alpha 8

Figure 5: Confusion Matrix

The generalization capability of the trained models was evaluated using a held-out test set composed of unseen RRC and NAS message sequences. As presented in Table 4, the best test accuracy of 84.33 percent was achieved by the rank 64 model, which also delivered the highest F1 score of 86.5 percent, with a precision of 90.6 percent and recall of 82.8 percent. The corresponding confusion matrix, shown in Figure 5a, illustrates this configuration's strong ability to correctly classify both benign and malicious signaling messages while keeping misclassifications low.

The rank 32 model also demonstrated robust performance, achieving a test accuracy of 82.84 percent, with a precision of 92.1 percent and the lowest false positive rate of 10.48 percent among all tested models. As illustrated in Figure 5b, this model was more conservative in its positive predictions, prioritizing the reduction of false alarms. These characteristics are particularly valuable in production environments, where incorrect alerts may degrade service quality or lead to unnecessary handover blocking.

5.2 Discussion

The results of this study highlight the effectiveness of using a unified SLM that processes both RRC and NAS signaling data simultaneously for detecting FBS. Unlike prior research that trains separate models for RRC and NAS features and later aggregates their outputs, our approach employs a single compact model trained on a joint representation. This design simplifies the detection pipeline and enables the model to capture deeper signaling patterns that span across both layers. The use of a unified architecture also reduces model management complexity and inference latency, which are critical in real-time, edge-deployed security systems.

RRC and NAS layers serve different roles in the cellular protocol stack. RRC is primarily responsible for radio-level operations such as connection establishment and mobility, while NAS handles higher-level functions like session management and authentication. Because of their distinct purposes, the feature sets extracted from RRC and NAS messages differ significantly in structure, type, and granularity. Previous studies have treated this difference as a challenge, choosing to build separate models for each layer to avoid feature incompatibility. In contrast, our method leverages textual transformation to represent all features uniformly as structured token-value pairs, allowing the SLM to process them within a single input stream.

By incorporating both RRC and NAS features into the same model, we enable the SLM to learn inter-layer dependencies that are often critical for accurate FBS detection. Sophisticated FBS attacks may attempt to mimic legitimate behavior at one layer while manipulating the other. For instance, a fake base station may clone RRC configuration parameters while subtly altering NAS reject causes or authentication flows. A model trained on a single layer might miss these cross-layer inconsistencies. However, our unified approach allows the model to detect such discrepancies by understanding the full signaling sequence and the contextual relationship between radio and protocol layers.

This joint modeling strategy is further validated by the strong performance of the fine-tuned models. The rank 64 configuration achieved high recall, suitable for high-sensitivity applications, while the rank 32 configuration offered high precision and the lowest false positive rate, ideal for minimizing unnecessary alerts. The confusion matrices presented in Figure 5 support these results, demonstrating that the unified model correctly classifies both benign and malicious signaling events with high consistency. These findings indicate that combining RRC and NAS data not only improves detection accuracy but also enhances the model's robustness against a wide range of attack strategies. Therefore, the proposed method represents a significant step forward in the design of lightweight, on-device FBS detection systems for future 6G networks.

6 Conclusion

This study presented a novel approach for FBS detection using a unified SLM fine-tuned on both RRC and NAS signaling data. By transforming structured signaling features into a textual representation, we enabled the Gemma3-270M-Instruct model to interpret and learn contextual patterns across protocol layers. Unlike prior methods that required separate models for RRC and NAS features, our unified model captures cross-layer dependencies within a single architecture, improving detection performance while reducing system complexity.

Experimental results demonstrated that the proposed method achieves high accuracy, precision, and recall across multiple configurations. The model with rank 64 offered high recall, making it suitable for aggressive detection scenarios, while the rank 32 model achieved the best precision and lowest false positive rate, ideal for conservative deployments. The use of both

RRC and NAS layers contributed significantly to the model's robustness, allowing it to detect a broad spectrum of FBS behaviors, including sophisticated and evasive attacks.

Overall, this work highlights the potential of compact, domain-adapted language models in securing 6G networks. The SLM-based approach offers a lightweight, scalable, and real-time solution for agentic AI at the network edge, reducing reliance on centralized infrastructure. Future work will explore further optimization techniques, such as quantization and federated fine-tuning, and extend the model to detect additional control-plane threats beyond FBS activity.

Acknowledgments

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIP) (No. RS-2024-00437252, Development of anti-sniffing technology for mobile communication and AirGap environments).

References

[1] The Pattaya News. Two Chinese Nationals Arrested for SMS Scam Using False Base Station in Bangkok, January 2025. Accessed: July 6, 2025.

[2] Khaosod English. Tour Guides Arrested in Bangkok's Sophisticated SMS Scam, January 2025. Accessed: July 6, 2025.

[3] Commsrisk. SMS Blaster Smishing Arrests in the UK, Qatar and Indonesia, June 2025. Accessed:July 6, 2025.

[4] Prajwol Kumar Nakarmi, Mehmet Akif Ersoy, Elif Ustundag Soykan, and Karl Norrman. Murat: Multi-RAT False Base Station Detector, February 2021.

[5] Zhou Zhuang, Xiaoyu Ji, Taimin Zhang, Juchuan Zhang, Wenyuan Xu, Zhenhua Li, and Yunhao Liu. FBSleuth: Fake Base Station Forensics via Radio Frequency Fingerprinting. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, ASIACCS '18, pages 261–272. Association for Computing Machinery.

[6] Jan Bolcek, Jan Kufa, Michal Harvanek, Ladislav Polak, Jan Kral, and Roman Marsalek. Deep Learning-Based Radio Frequency Identification of False Base Stations. In *2023 Workshop on Microwave Theory and Technology in Wireless Communications (MTTW)*, pages 45–49.

[7] Hoonyong Park, Philip Virgil Berrer Astillo, Yongho Ko, Yeongshin Park, Taeguen Kim, and Ilsun You. SMDFbs: Specification-Based Misbehavior Detection for False Base Stations. 23(23):9504.

[8] Ankush Singla, Rouzbeh Behnia, Syed Rafiul Hussain, Attila Yavuz, and Elisa Bertino. Look Before You Leap: Secure Connection Bootstrapping for 5G Networks to Defend Against Fake Base-Stations. In *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security*, ASIA CCS '21, pages 501–515. Association for Computing Machinery.

[9] Prajwol Kumar Nakarmi, Jakob Sternby, and Ikram Ullah. Applying Machine Learning on RSRP-based Features for False Base Station Detection. In *Proceedings of the 17th International Conference on Availability, Reliability and Security*, pages 1–7.

[10] Daehyeon Son, Youngshin Park, Bonam Kim, and Ilsun You. A Study on the Implementation of a Network Function for Real-time False Base Station Detection for the Next Generation Mobile Communication Environment. 15(1):184–201.

[11] Kazi Samin Mubasshir, Imtiaz Karim, and Elisa Bertino. Gotta Detect 'Em All: Fake Base Station and Multi-Step Attack Detection in Cellular Networks.

[12] Peter Belcak, Greg Heinrich, Shizhe Diao, Yonggan Fu, Xin Dong, Saurav Muralidharan, Yingyan Celine Lin, and Pavlo Molchanov. Small Language Models are the Future of Agentic AI.

[13] Introducing Gemma 3 270M: The compact model for hyper-efficient AI- Google Developers Blog.

[14] Qwen Team. Qwen3: Think Deeper, Act Faster.