A Method for Detecting Out-of-Distribution Examples in Encrypted Mobile Traffic Classification

Abstract—**The widespread use of encrypted communication in mobile networks poses significant challenges in accurately classifying traffic. Detecting out-of-distribution (OOD) examples, which significantly deviate from known classes, adds complexity to the task. In this paper, we propose a feature analysis-based OOD detection scheme for traffic classification in Long-Term Evolution (LTE) systems. Our method utilizes Long Short-Term Memory (LSTM) networks for feature extraction, capturing the feature vectors of the traffic series. Principal Component Analysis (PCA) is then applied to obtain principal and residual principal components. Leveraging the residual feature vector, we construct an OOD score to quantify deviation from the ID dataset. Extensive experiments on a large-scale encrypted mobile traffic dataset demonstrate the superiority of our approach, achieving high accuracy in OOD detection compared to existing techniques. Our method contributes to enhanced security and reliable traffic classification in LTE systems, addressing challenges posed by OOD examples.**

Index Terms—**Encrypted mobile traffic, Long-Term Evolution (LTE), Out-of-distribution detection, Traffic classification**

I. Introduction

In modern mobile communication systems, traffic classification is an essential issue related to security and the quality of service (QoS) [1]. Due to the encryption used in LTE and 5G networks, there is no method for passively acquiring traffic data from the network or transport layer. However, with the help of the LTE downlink decoding software (e.g., SRS Airscope [2] or OWL [3]), analyzing the Downlink Control Information (DCI) becomes possible. In LTE systems, DCI messages are transmitted in the Physical Downlink Control Channel (PDCCH) and contain various control information, such as modulation and coding schemes, resource allocation, and power control commands. Utilizing the traffic information from DCI, traffic classification with a passive sniffer becomes feasible. Various methods and techniques have been explored for using DCI to categorize network traffic [4]–[6].

Although these methods achieve good performance in services or app usage classification by utilizing machine learning methods, the effectiveness of these models is based on a close-set assumption. That is to say, the training and testing samples are from the same dataset. However, for practical use, traffic classification is often performed in an open-world setting, which means large amounts of traffic data generated by mobile apps have never been trained by the machine learning model. The traffic data generated by these untrained apps can be viewed as out-of-distribution (OOD) samples. A classifier must effectively detect the OOD samples to have practical usages in real-world data.

Some method has been used to address the traffic classification OOD detection problem. Thijs van Ede *et al.* reported using Jaccard similarity to detect unseen apps from the training data [7]. However, this method is hard to be applied in deep learning methods. In [5], the authors leveraged kernelized spatial depth function [8] to estimate the similarity between the testing and training samples and perform OOD detection. However, the authors default to the method's validity and do not evaluate its performance. Another work reported by Madushi H. Pathmaperuma [9] introduces a method deploying a Deep Neural Network (DNN) to classify different apps by setting a threshold for the maximum softmax probabilities(MSP). However, the methods based on MSP or logits, as described in [10] and [11], respectively, may encounter a challenge in accurately detecting out-of-distribution (OOD) data in traffic applications. This is because they assume that the test data follows the same distribution as the training data, and in-distribution (ID) data will have high MSP. It is important to note that a large softmax score of a particular class does not always reflect a high similarity between the test sample and that class. In some cases, a particular class may be assigned a high softmax score simply because the test sample is significantly dissimilar to all other classes.

To address this issue and perform reliable out-of-distribution (OOD) detection, we propose to extract features from the traffic series and use them for OOD detection. Specifically, we apply Principal Component Analysis (PCA) to analyze the features extracted from the training data. We assume the OOD samples will project more onto the residual component than the in-distribution samples. This is because the residual component captures the unimportant part of the data. By projecting the features onto the residual matrix of the PCA, we can effectively detect OOD samples and avoid the influence of overconfident logit or softmax scores.

In this paper, we present an effective method for OOD detection in encrypted traffic classification to fill the blank in this topic. We briefly summarize our contributions as follows:

- We propose an OOD detection scheme for mobile encrypted traffic classification based on feature analysis in combination with a Long Short-Term Memory (LSTM) [12] model.
- We collect a dataset with real-world LTE traffic data consisting of 20 mobile apps for training and testing our model. The experiment results demonstrate that our

[*] These authors contributed equally to this work.

method has high detection effectiveness and outperforms the baselines.
- We analyze the impact of the hyperparameters we set on the different OOD datasets.

II. Methodology

A. Problem Setting

This paper investigates the problem of distinguishing traffic series of ID and OOD apps on a pre-trained LSTM network. Since all the collected data is from the LTE system, we can assume that all the apps share the same feature space. In contrast, different apps have various feature distributions corresponding to their functionalities. Formally, we consider two domains $\mathcal{D}_{in} = \{\mathcal{X}, P_{in}(X)\}$, and $\mathcal{D}_{out} = \{\mathcal{X}, P_{out}(X)\}$, where $\mathcal{X}$ is a feature space, $P_{in}(X)$ and $P_{out}(X)$ are marginal probability distributions of ID and OOD apps' feature vectors respectively, formally, $P(X) = \Pr\{x = \boldsymbol{x}_i\}, \boldsymbol{x}_i \in \mathcal{X}$, each x_i corresponds to a specific feature vector. Our model has two tasks: (1) classify the ID apps; (2) detect the OOD apps.

The first task can be defined as $\mathcal{T} = \{\mathcal{Y}, f(.)\}$, where $\mathcal{Y} = \{y_1, y_2, \ldots, y_m\}$ is the label space and $f(.)$ is the objective predictive function $f : \mathcal{X} \rightarrow \mathcal{Y}$. We train an LSTM model with the training set D_{train} consisting of ID apps to complete this task. The model is defined as follows:

$$f(.) = f_{FC}(f_{LSTM}(.)) \tag{1}$$

where $f_{FC}(.)$ and $f_{LSTM}(.)$ are the fully connected layer and LSTM layers.

We use the notation from [13] to describe the second task. The test set includes traffic series drawn from both $P_{in}(X)$ and $P_{out}(X)$. We can consider that the testing samples follow a joint distribution $\mathbb{P}(X, Z)$ defined on $\mathcal{X} \times \{0, 1\}$, where the conditional probability distributions satisfy $\mathbb{P}(X|Z = 1) = P_{in}(X)$ and $\mathbb{P}(X|Z = 0) = P_{out}(X)$. Then the problem is that the model should distinguish whether a sample drawn from $\mathbb{P}(X, Z)$ is an ID or OOD sample. In this paper, we mainly focus on the second task since traffic classification can be effectively performed once the sample is reported as an ID sample.

B. Model

We propose a method based on feature analysis to perform effective OOD detection. It is generally accepted that the significant features of a sample lie in low-dimensional manifolds [14]. Thus, we can perform PCA to decompose the entire feature space $\mathcal{X}$ into a principal space P and a residual space $P^{\perp}$. A feature vector deviating from the principal space is likely to indicate an OOD sample. To measure the deviation from the principal space, we define the residual of the feature vector as:

$$\text{Residual}(\boldsymbol{x}) = ||\boldsymbol{x}^{P^{\perp}}|| \tag{2}$$

where $x^{\perp}$ is the projection of x to the residual space. And we use the residual as the OOD score to determine whether a sample is OOD or ID. The overall procedure of our scheme is as follows.

- We first train the LSTM $f_{LSTM}(.)$ using the ID dataset D_{train}. We keep the feature vectors $\boldsymbol{x}_i, i = 1, ..., |D_{train}|$ of all the ID samples during the training process. Define $\mathbf{X} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_{D_{in}})^T$ as the feature matrix of the entire ID dataset and calculate its covariance matrix $Cov(\mathbf{X})$.
- Perform the PCA and obatin an L-dimensional subspace spanned by the eigenvectors with the largest L eigenvalues of the covariance matrix $Cov(\mathbf{X})$. We define this L-dimensional space as the principal space P. The residual space $P^{\perp}$ is the orthogonal space spanned by the remaining eigenvectors. The hyper parameter L is tuned by setting the cumulative explained variance $\gamma = \frac{\sum_i^L \lambda_i}{\sum_j^N \lambda_j}$, where λ_i are eigenvalues of $Cov(\mathbf{X})$ arranged increasingly, and N is the number of features.
- To calculate the threshold δ, we first project the feature vector $\boldsymbol{x}_i$ to the principal space and obtain a lower dimensional feature vector $\boldsymbol{x}_i^{P^{\perp}} = Q_{N-L}x_i$, where $Q_L = (q_{L+1}, ..., q_N)^T$, and $\{q_i\}_{i=L+1}^N$ are the eigenvectors spanning the residual space. A commonly-used threshold δ is a value making 95% of the ID samples correctly classified.
- During testing, we extract the feature vector $\boldsymbol{x}_j, j = 1, ..., |D_{test}|$ form test set D_{test}. Project the feature vector $\boldsymbol{x_j}$ of the testing data to the residual space and obtain a lower dimensional feature vector $\boldsymbol{x}_j^{P^{\perp}} = Q_L x_j$. Calculate the residual $||\boldsymbol{x}_j^{P^{\perp}}||$ as the OOD score and compare it with the threshold δ to determine whether the sample is ID or OOD. Mathematically, the out-of-distribution detector can be described as:

$$g(\boldsymbol{x}, \delta) = \begin{cases} 1 & \text{if } ||\boldsymbol{x}^{P^{\perp}}|| \leq \delta, \\ 0 & \text{if } ||\boldsymbol{x}^{P^{\perp}}|| > \delta \end{cases} \tag{3}$$

where $g(\boldsymbol{x}, \delta) = 1$ indicates the sample is ID while $g(\boldsymbol{x}, \delta) = 0$ indicates the sample is OOD.

III. Experiment and Analysis

We first introduce how to collect and process the traffic data of different apps in the LTE system in the first two sections. Then, we demonstrate the experiment process for validating the proposed method and hypothesis. Here we conclude the three hypotheses we made in this paper:

- OOD samples have larger residual components than ID samples.
- OOD apps providing similar functionalities and services with ID apps are more likely to be misclassified.
- The residual-based method is less vulnerable to the similarity between OOD and ID samples.

A. LTE Identifier

The identifiers in the LTE network include Mobile Subscriber Identity (IMSI), Temporary Mobile. Subscriber Identifier (TMSI), and Cell Radio Network Temporary Identifier (C-RNTI). TMSI is used to identify a SIM card globally. For security, when the subscriber initiates a connection to the LTE network, the LTE system assigns a TMSI to the subscriber.

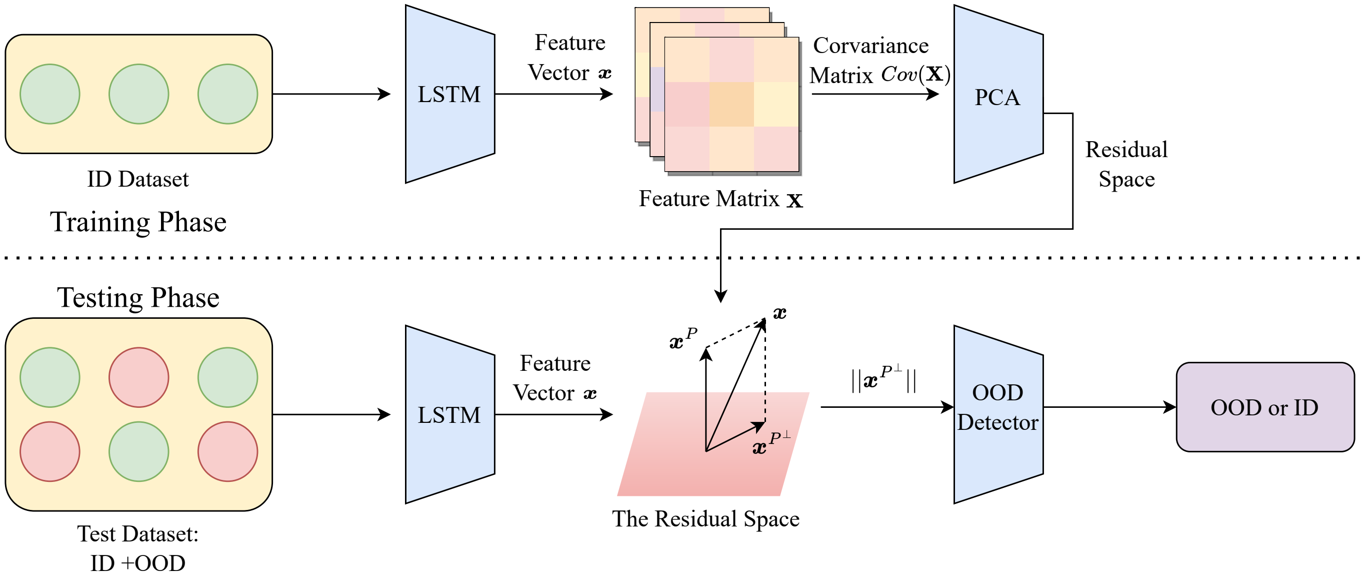

ID Dataset
Training Phase
LSTM
Feature Vector $\boldsymbol{x}$
Feature Matrix $\mathbf{X}$
Corvariance Matrix $Cov(\mathbf{X})$
PCA
Residual Space
Testing Phase
Test Dataset: ID +OOD
LSTM
Feature Vector $\boldsymbol{x}$
$\boldsymbol{x}^P$
$\boldsymbol{x}$
$\boldsymbol{x}^{P^\perp}$
The Residual Space
$||\boldsymbol{x}^{P^\perp}||$
OOD Detector
OOD or ID

Fig. 1. The OOD detection scheme for the LTE traffic classification system

When setting up the Radio Resource Control (RRC) connection, eNodeB broadcast the TMSI to the subscriber without encryption [15]. Although TMSI is not an unchangeable identifier, the refreshing rate is much lower than C-RNTI. Thus, TMSI enables us to monitor specific user equipment for a relatively long time.

B. Data acquisition

To acquire traffic data of different apps, we first set up a connection between the experiment user equipment (UE) and the base stations, Evolved NodeB (eNodeB). To monitor the traffic generated by UE, we use the SDR, USRP X310, and the LTE downlink decoding software, SRS Airscope. We set the operating frequency of the SDR as the same as the eNodeB connecting with UE so that the SDR can receive the broadcasting control packets from PDCCH. AirScope can decode the packets from PDCCH and present DCI messages, including the RRC connection setup message. The deployment positions of the SDR, UE, and eNodeB are shown in Fig. 2. One crucial issue is to specify UE and filter out the UE-related DCI messages. To address this issue, we first uploaded a large file to create an observable burst uplink data rate to determine the initial RNTI for UE. Then we set UE to use specific mobile apps. To form a comprehensive dataset for LSTM training and OOD detection, we selected 20 apps providing 5 common service types. We select 5 apps with different service types as the in-distribution (ID) classes. We collect 100 traffic traces for each ID class,80 for training, and 20 for testing. The remaining 15 apps are arranged into 3 groups of OOD test sets. Each OOD class has 20 traffic traces. Each trace is a 20-second traffic series. The list of ID apps is shown in Table I, while the grouping of OOD apps is shown in Table II. Afterward, multiple RNTIs are mapped with TMSI. By using these RNTIs, the corresponding DCI messages for UE are filtered out. This paper uses the Transport Block Size (TBS) in DCI messages as the raw data for traffic analysis.

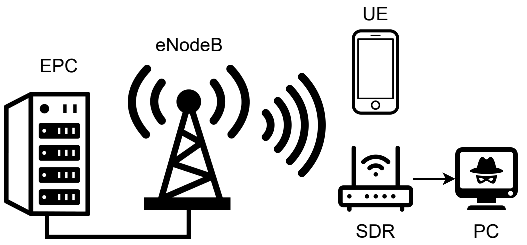


Fig. 2. LTE Network Infrastructure and the position of the traffic monitor

TABLE I
NAMES AND CATEGORIES OF THE ID APPS

Traffic service type	**Representative app**
Streaming Video	TikTok
Streaming Music	Spotify
Social Media	Instagram
Text Chat	Whatsapp
Video Calls	Whatsapp

C. Experiments and results

In order to evaluate the proposed scheme, we compare the performance of our method with two baselines [10], [11].

- Energy-based detection uses logits of a sample to calculate the energy function $E(x)$ as the OOD score. The energy function is given by:

$$E(x) = -T \cdot \log \sum_{i}^{K} e^{(f_{FC}^{i}(x)/T)}$$

TABLE II
GROUPING OF THE OOD APPS

Set 1	Set 2	Set 3
Pubg Mobile	Google	Xiaohongshu
Shopee	YouTube	Wechat Text
Grab	Amazon Prime Video	Telegram Text
Genshin	Netflix	Wechat Video
Lazada	YouTube Music	Telegram Video

where T is a scalar as the temperature parameter, $f^i_{FC}(x)$ is the logit of the ith class, and K is the number of classes.

- Softmax-based detection uses the largest softmax score as the ID score. Unlike the OOD score, a larger ID score indicates that a sample is more likely to be OOD.

All the methods use the same pre-trained LSTM model and are evaluated among three test sets consisting of ID data in Table I and OOD data shown in Table II. The ID training set includes 5 classes, each with 80 samples. Each testing set consists of 5 ID and 5 OOD classes, with 20 samples per class. When constructing the test sets, we set different OOD detection difficulties by controlling the similarity between OOD and ID apps. Set 1 consists of apps with significantly different functionalities than the training set. Thus, the OOD detection for set 1 should be the easiest. Set 2 consists of apps providing somewhat similar functionalities to the training apps. However, the streaming video apps in set 2 have different service preferences than those in the training set. TikTok is an app for short videos; YouTube focuses more on long videos, while Amazon Prime Video and Netflix provide TV series and movies. Since apps in set 2 have similar functionalities to the training apps, OOD detection becomes more difficult. Set 3 includes one social media app and two instant messaging apps providing text chats or video calls. Set 3 consists of apps having very similar services and functionalities to apps in the training set. And it makes set 3 have the greatest OOD detection difficulty. We use these intentionally designed test sets to verify our second and third hypotheses.

The experiment results are shown in Table III. All values are in percentages. ↑ indicates higher value for better performance, while ↓ indicates lower value for better performance. We set the hyperparameter $\gamma = 0.999$, resulting in principal feature space with $L = 62$ dimensions. We consider three commonly-used metrics to evaluate each OOD method's performance.

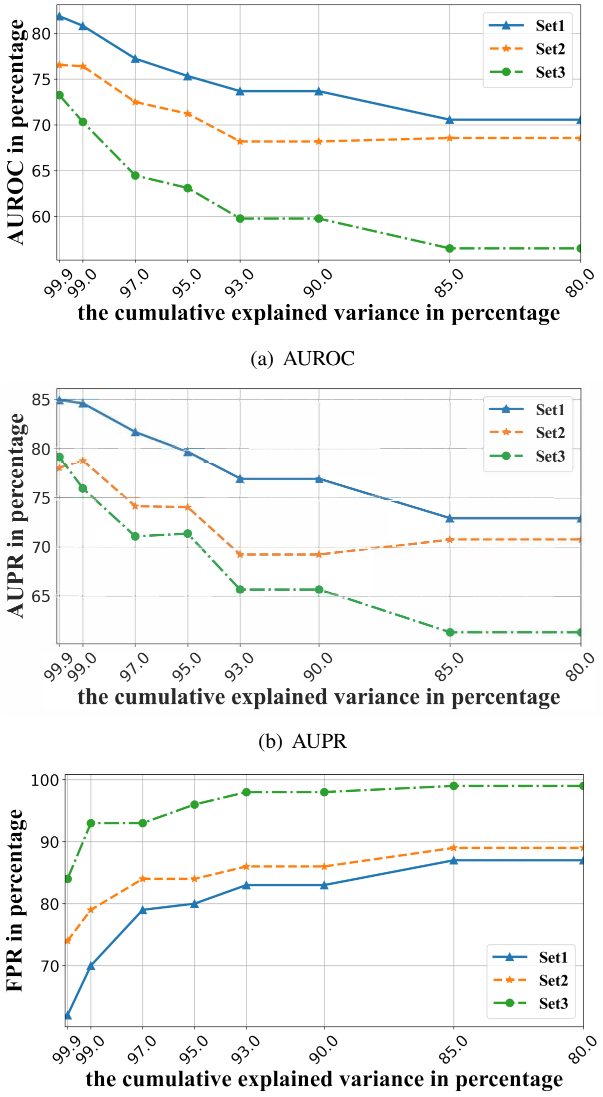
- **FPR95** is the false positive rate (FPR) when the true positive rate (TPR) achieves 95%. FPR is the proportion of OOD samples misclassified as ID. TPR is the proportion of correctly classified ID samples.
- **AUROC** is the Area Under the Receiver Operating Characteristic curve, a threshold-independent metric proposed in [16]. It is defined as the area under the curve obtained by plotting the True Positive Rate (TPR) against the False Positive Rate (FPR) at various classification thresholds. AUROC interprets the probability that an ID sample has a higher score than an OOD sample [17]. A perfect OOD detector has an AUROC score of 100%.
- **AUPR** is the Area under the Precision-Recall curve, which is also a threshold-independent metric [18]. Precision is defined as TP/(TP+FP), and recall refers to TP. Similarly, a perfect OOD detector has an ARPR score of 100%.

From Table. 3, the proposed method outperforms the baselines in all three testing sets. Taking an average of each metric over the three sets, our proposed method has 14.74% and 19.38% higher AUROC, 15.34%, and 16.86% higher AUPR, and 13% and 16% lower FPR than the softmax-based and energy-based methods correspondingly. The results validate the effectiveness of our proposed method and correspond with our first hypothesis. The AUROC measures the probability that an OOD sample has a larger OOD score than an ID sample. And for the residual-based method, the OOD score is the Euclidean norm of the residual feature vector. Thus, the experiment results demonstrate that an OOD sample is very likely to have a larger residual component than an ID sample.

Meanwhile, we observe that the effectiveness of each method degrades as the similarity between ID and OOD apps increases. The observation is consistent with common sense since we assume that apps providing similar services have similar traffic patterns and therefore have similar feature vectors. Consequently, an OOD app may be misclassified as an ID app with similar functionalities or services. The second hypothesis is verified.

The experiment results also indicate that the proposed method is more robust than the logits-based or energy-based methods. Even in set 3, consisting of apps highly similar to the training ones, the residual-based method still has 17.29% and 25.96% higher AUROC, 16.55% and 21.15% higher AUPR, and 12% and 14% lower FPR than the softmax-based and energy-based methods correspondingly. These results show our method's effectiveness in different testing environments and verify the third hypothesis. According to [19], logits and softmax scores only consider class-dependent information, which is why they are more vulnerable to class-agnostic information. Class-dependent information is the unique characteristic of each class, while class-agnostic information refers to the common statistical properties and patterns across all classes. The residual-based method pays less attention to the class-dependent information, emphasizing the similarity between the testing sample and the training set. This characteristic gives the residual-based method more robustness against the class-agonist information. And that is the reason why it can perform effectively even in severe environments like set 3.

We perform ablation experiments to examine the impact of PCA in our scheme in the three test sets, and the results are shown in Fig. 3. We tune the cumulative explained variance γ to have a different decomposition of the feature space and find an optimal value partitioning the principal and residual spaces. The experiment results show that, in SET 1 and SET 3, the metrics are monotonic to γ, which indicates that a larger γ value provides better performance to our scheme. In set 2, as the γ decreases, the AUROC decreases and then slightly



(a) AUROC

(b) AUPR

(c) FPR

Fig. 3. The metrics of different sets w.r.t the cumulative explained variance in percentage

increases. The AUPR has a similar tendency to the AUROC, but the only difference is that the highest AUPR is achieved by $\gamma = 99\%$. And the FPR of set 2 is also monotonic to γ. From these observations, we can conclude that choosing $\gamma = 99.9\%$ provides the best performance considering all the sets and metrics. Another observation is that the gradient of the curves becomes smaller as the γ increases. This observation indicates that the partitioning of the principal and residual spaces is subtle. An eigenvector with a small eigenvalue may be an essential principal or residual space basis. And a slight change in the feature space partitioning could significantly influence the performance of the entire scheme.

IV. Conclusion

In this paper, we introduce an OOD detection scheme for LTE traffic classification. The scheme utilizes feature analysis and dimension reduction techniques for effective OOD detection. The main idea is to decompose the entire feature space into two subspaces and project the feature vector to the residual subspace. We use the Euclidean norm of the residual feature vector as the OOD score to decide whether the sample is ID or OOD. To evaluate the effectiveness of our method, we collected a real-world LTE traffic dataset. The experiment results indicate that the proposed scheme significantly outperforms the baseline methods. And we also demonstrate the impact of PCA in our scheme with a series of ablation experiments.

References

[1] I. L. Cherif and A. Kortebi, “On using extreme gradient boosting (xgboost) machine learning algorithm for home network traffic classification,” in *2019 Wireless Days (WD)*. IEEE, 2019, pp. 1–6.

[2] Software Radio Systems, “SRS AirScope,” [Accessed on Oct. 30, 2022]. [Online]. Available: https://www.srs.io/products/

[3] N. Bui and J. Widmer, “OWL: A reliable online watcher for LTE control channel measurements,” in *Proceedings of the 5th Workshop on All Things Cellular: Operations, Applications and Challenges*, 2016, pp. 25–30.

[4] J.-W. Son, S. Lee, and M.-h. Han, “Supervised Service Classification using Downlink Control Indicator in LTE Physical Downlink Control Channel,” in *2021 International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE, 2021, pp. 1533–1536.

[5] H. D. Trinh, A. F. Gambin, L. Giupponi, M. Rossi, and P. Dini, “Mobile traffic classification through physical control channel fingerprinting: a deep learning approach,” *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1946–1961, 2020.

[6] L. Zhai, Z. Qiao, Z. Wang, and D. Wei, “Identify What You are Doing: Smartphone Apps Fingerprinting on Cellular Network Traffic,” in *2021 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2021, pp. 1–7.

[7] T. Van Ede, R. Bortolameotti, A. Continella, J. Ren, D. J. Dubois, M. Lindorfer, D. Choffnes, M. van Steen, and A. Peter, “Flowprint: Semi-supervised mobile-app fingerprinting on encrypted network traffic,” in *Network and Distributed System Security Symposium (NDSS)*, vol. 27, 2020.

[8] Y. Chen, X. Dang, H. Peng, and H. Bart, “Outlier detection with the kernelized spatial depth function,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 288–305, 2009.

[9] M. H. Pathmaperuma, Y. Rahulamathavan, S. Dogan, and A. M. Kondoz, “Deep learning for encrypted traffic classification and unknown data detection,” *Sensors*, vol. 22, no. 19, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/19/7643

[10] D. Hendrycks and K. Gimpel, “A baseline for detecting misclassified and out-of-distribution examples in neural networks,” *arXiv preprint arXiv:1610.02136*, 2016.

[11] W. Liu, X. Wang, J. Owens, and Y. Li, “Energy-based out-of-distribution detection,” *Advances in neural information processing systems*, vol. 33, pp. 21 464–21 475, 2020.

[12] F. A. Gers, J. Schmidhuber, and F. Cummins, “Learning to forget: Continual prediction with lstm,” *Neural computation*, vol. 12, no. 10, pp. 2451–2471, 2000.

[13] S. Liang, Y. Li, and R. Srikant, “Enhancing the reliability of out-of-distribution image detection in neural networks,” *arXiv preprint arXiv:1706.02690*, 2017.

[14] A. Zaeemzadeh, N. Bisagno, Z. Sambugaro, N. Conci, N. Rahnavard, and M. Shah, “Out-of-distribution detection using union of 1-dimensional subspaces,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 9452–9461.

[15] D. Rupprecht, K. Kohls, T. Holz, and C. Pöpper, “Breaking lte on layer two,” in *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2019, pp. 1121–1136.

[16] J. Davis and M. Goadrich, “The relationship between precision-recall and roc curves,” in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 233–240.

TABLE III
OOD DETECTION FOR THE PROPOSED AND BASELINE METHODS

OOD	**Softmax**			**Energy**			**Ours**		
Dataset	AUROC↑	AUPR↑	FPR95↓	AUROC↑	AUPR↑	FPR95↓	AUROC↑	AUPR↑	FPR95↓
Set 1	67.16	69.53	77.00	58.56	60.62	82.00	81.87	84.94	62.00
Set 2	64.31	63.97	86.00	67.78	72.92	88.00	76.55	78.03	74.00
Set 3	55.96	62.60	96.00	47.29	58.00	98.00	73.25	79.15	84.00
Average	62.48	65.37	86.33	57.84	63.85	89.33	77.22	80.71	73.33

[17] T. Fawcett, “An introduction to roc analysis,” *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.

[18] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, “Universal adversarial perturbations,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1765–1773.

[19] H. Wang, Z. Li, L. Feng, and W. Zhang, “Vim: Out-of-distribution with virtual-logit matching,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.