Cybersecurity in Public Space: Leveraging CNN and LSTM for Proactive Multivariate Time Series Classification

Aimen Ahmed AL Odaini
University of Cassino and Southern Lazio
03043 Cassino, Italy
aalodaini@vicomtech.org

Francesco Zola
Vicomtech, Basque Research and Technology Alliance (BRTA)
20009 Donostia/San Sebastian, Spain
fzola@vicomtech.org

Lander Segurola-Gil
Vicomtech, Basque Research and Technology Alliance (BRTA)
20009 Donostia/San Sebastian, Spain
lsegurola@vicomtech.org

Amaia Gil-Lertxundi
Vicomtech, Basque Research and Technology Alliance (BRTA)
20009 Donostia/San Sebastian, Spain
agil@vicomtech.org

Carmen D'Andrea
University of Cassino and Southern Lazio
03043 Cassino, Italy
carmen.dandrea@unicas.it

2023 IEEE International Conference on Big Data (BigData) | 979-8-3503-2445-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/BigData59044.2023.10386165

***Abstract*—Mobile communications have become a vital domain for criminals and terrorists to exploit vulnerabilities, posing significant threats to public safety and national security. More precisely, they can employ cell site simulators such as International Mobile Subscriber Identity (IMSI) catchers to intercept and monitor mobile communications, enabling eavesdropping, tracking individuals' movements, and potentially coordinating illegal activities while evading detection by law enforcement agencies. To overcome this issue, an innovative approach to detect IMSI catchers using Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) models is proposed. Leveraging the power of deep learning, the developed models process multivariate time series data to distinguish suspicious patterns indicative of IMSI catcher presence. This work compares nine different model architectures based on CNN, LSTM, or a combination of both through a series of case studies. We demonstrate that LSTM-based and Parallel CNN/LSTM models outperform other architectures, achieving high precision and recall rates. Then, the best two models are tested with several sequence lengths. The presented models serve as valuable tools, providing a further enhancement to the security of mobile networks. The goal of this research work is to contribute to the broader mission of integrating artificial intelligence within the daily investigative practices of law enforcement agencies.**

***Index Terms*—CNN, LSTM, Multivariate Time Series, Suspicious Behaviour Classification, Cybersecurity in Public Space**

I. Introduction

Nowadays, the massive paradigm shifts in information technologies and digital infrastructure, as well as new paradigms like Smart Cities and Homes, Edge computing, and 5G, have created a more intelligent environment where even more devices are interconnected. In that sense, devices ranging from home assistants and smartwatches to drones and cars, but also traffic lights, fridges, and any sensor, continually amass, process, and store data to enhance user experiences and facilitate their day-to-day tasks. This new scenario has resulted in a major surface of cyberattacks, i.e., critical points that can suffer from a high-impact attack [1], [2], [3]. At the same time, the number of illegal activities has increased since this new information can be used as a facilitator for performing many cybercrimes [4].

To respond to this escalation, law enforcement agencies (LEAs) have started to apply paradigms like Big Data and Artificial Intelligence (AI) to enhance the security of the public space by promptly detecting dangerous/suspicious activities that can be related to criminal investigation and even terrorism activities [5]. Thereby, they started to acquire expertise in using AI tools for analyzing and correlating huge amounts of data, events, transactions, videos, images, etc. [6], [7]. In fact, these AI solutions not only can be used to enhance their investigative capabilities but also to enable them to predict when and where new incidents are most likely to occur. Yet, AI offers LEAs the potential to optimize productivity and efficiency by streamlining processes, identifying patterns, detecting threats at early stages, and making faster and more accurate decisions. However, to maximize the benefit of AI, LEAs must take a critical and human-centric approach to implementing AI technologies, prioritizing the safety and privacy of society [8].

One of the primary sources of personal threat data collection is still represented by the mobile network. According to Kaspersky[1], data leakage, unsecured Wi-Fi, spoofing, phishing, spyware, and broken cryptography are among the most common mobile threats [9]. In particular, it is possible to buy devices that allow attackers/hackers to eavesdrop on calls and texts, install spyware on the mobile, jam the traffic or simply

[1]https://www.kaspersky.com

take control of the mobile [10]. IMSI catcher represents an example of this. An IMSI-catcher is a surveillance device used to intercept and monitor mobile communications in a specific area by mimicking a legitimate cell tower. This trick enables hackers to capture various types of information from connected devices [11]. However, while such valuable data are usually collected by LEAs during their investigations to address threats and crimes in specific areas, our case is the opposite: when criminals use IMSI catchers for unlawful activities. In this scenario, with the aid of AI, LEAs can identify and predict suspicious activities and attacks that are timely associated with IMSI catcher utilization.

In this context, LEAs must analyze the attributes that indicate the presence of an IMSI catcher in a mobile network, considering their temporal dependence, i.e., modelling them as a time series. In that way, it will also be possible to correlate the temporality and the possible periodicity of specific crimes. For this reason, in this paper, we propose to use deep learning (DL) techniques for classifying suspicious behaviours associated with IMSI catchers. More specifically, we compare the benefits and limitations of two architectures that have already shown potential in multivariate time series classification [12]: CNN and LSTM. CNNs have the ability to automatically identify and extract important patterns from the input time series data using convolution and pooling operations [13]. While on the other hand, LSTMs are adept at modeling short- and long-term dependencies in sequential data using the gating mechanism and the built-in memory cells. Firstly, we study how the two separate structures perform the multivariate time series classification task. Then, we combine them in three different architectures: i) cascading layers with CNN and then LSTM here called CNN+LSTM, ii) cascading layers inverting the order here called LSTM+CNN, and iii) using them in parallel in the architecture called LSTM/CNN. Experiments validate how different suspicious levels of the behaviours and history length, i.e., the number of elements considered at once as memory, affect the quality of the predictions. The results demonstrate the significant capability of LSTM-based models, whether used individually or in combination with CNN in a parallel structure. Moreover, these architectures tend to work better when dealing with shorter sequence lengths, which helps the model reduce the impact of the class imbalance issue within our dataset. This breakthrough paves the way for broader applications in LEA practices and investigations.

The paper is organized as follows: In Section II we provide a general background of the relevance of IMSI catcher detection process, CNN and LSTM networks, as well as related works. In Section III, the methodology proposed in this study is described, whereas in Section IV, the dataset, the experiments, the metrics used for their evaluation, and the architecture configurations are reported. Section V reports and discusses the results. Finally, Section VI concludes the paper and provides some guidelines for future work.

II. Preliminaries

This section is organized as follows: in Subsection IV-A, we establish a knowledge base by discussing the attributes that indicate the presence of an IMSI catcher in a mobile network, while Subsection II-B provides brief definitions of convolutional and temporal learning. Finally, we review related work in Subsection II-C.

A. IMSI Catcher in Mobile Networks

Detecting the presence of IMSI catchers in the mobile phone network represents a fundamental necessity, as they cause several threats and risks to public and cyberspace safety in various contexts. IMSI catchers are devices purposely built to intercept mobile communications, including voice calls, text messages, and data traffic, granting attackers unauthorized surveillance capabilities. They also extend to data theft, where IMSI catchers can capture sensitive data transmitted via mobile networks, including login credentials, personal details, and financial information, thereby creating opportunities for identity theft and fraudulent activities. Additionally, IMSI catchers can be used to track the location of mobile devices that are connected to them. Furthermore, these devices serve as potential entry points for cyberattacks, making connected devices susceptible to malware deployment and data breaches. These misuses can make the deployment of IMSI catchers by cyber criminals or terrorists pose national security risks, potentially targeting government agencies, military installations, critical infrastructure and disrupting mobile communication services. Disruptions during emergencies or critical events can create chaos, hinder response efforts, and amplify the impact of attacks. These threats encourage LEAs to adopt innovative AI countermeasures in response to these challenges.

The uses of DL models in detecting IMSI catcher centre on three main types of events indicating its presence. The first is related to a sequence of network anomalies, such as the network force down from 3G to 2G mode, the current cell has no neighbouring cell, and the connection is not encrypted. The second type of event is related to the baseband processor (BP) activity in correlation with the phone's application processor (AP), in other words, the activity of BP and AP should be timely correlated. The third type is related to signal strength, as IMSI catchers often emit stronger signals than legitimate cell towers.

B. CNN and LSTM Networks

Although CNNs are primarily designed for image-related tasks, they can also be adapted for time series data through two operations known as one-dimensional (1D) convolution and pooling, the first is done by applying a one-dimensional filter (also known as kernels) to the input sequences. This filter convolves as a sliding window that scans through the time series, identifying important deep features. Then in the second operation, extracted features are passed through pooling layers whose role is to extract the most important information from the output of the convolutional layer. Lastly, a fully connected

layer is applied to map the learned features from the sequential data to the final output [13].

On the other hand, LSTM is based on using a combination of memory cells and a series of gating mechanisms. It includes a cell state, representing LSTM's memory, and a hidden state, which carries relevant information from previous time steps to the current one. These memory cells interact with several gating mechanisms: the forget gate, which determines what information in the memory cell should be discarded; the input gate, which decides what new information should be added to the cell state; and finally, the output gate, which determines what information the cell state should reveal as the output. This combination improves gradient flow and enhances the network's ability to handle long-term dependencies and variable-length sequences. These architectural advancements and mechanisms make LSTMs highly effective in capturing complex patterns, modeling long-term dependencies, and learning from sequential data [14]

CNNs and LSTMs can form a powerful alliance for time series classification by leveraging their unique capabilities to learn the most discriminative features from raw data and recognize deep temporal patterns.

C. Related Work

With the rapid advancements in technology, AI has the potential to revolutionize traditional LEA practices by enhancing investigation processes, improving predictive capabilities, and increasing operational efficiency. From analyzing vast amounts of data to identifying patterns and suspicious events, several AI tools and solutions in different application areas have proven to be effective assets in the fight against traditional crime and cybercrime [15]. AI applications are already being used to help LEAs detect and prevent crimes. Elluri et al. [16] developed a crime prediction model to send high or low crime alerts to LEAs using DL algorithms, while Chun et al. [17] presented a neural network-based model to predict whether a specific person will commit a new crime in the future using an individual's criminal charge history records. Yet, Saraiva et al. [18] introduced a crime prediction and monitoring model that uses various data sources, including historical crime records, geographical information, and textual data from social media. Al-Khater et al. [19] conducted a comprehensive review of existing cybercrime detection techniques.

The majority of these works try to apply the knowledge gathered in other domains such as video surveillance [20], [21], face detection and recognition [22], [23], crypto-analysis [24] towards crime prediction and prevention for helping LEAs in their investigation. Usually, the process of detecting anomalous behaviours requires a complete temporal analysis of historical data to identify significant patterns or deviations. Consequently, LEAs need to adopt temporal-based AI solutions akin to those used in the cybersecurity domain, such as in [25], a malware detector using LSTM was built to extract behavioural features from time series data and CNN to classify the converted features to either normal or malicious. [26] Achieved great accuracy results by constructing a combined CNN and LSTM network for malware classification. In [27], the author proposed an intrusion detection system based on a hybrid network of CNN and LSTM. In [28], the author presented a phishing detection system using DL models with CNN, LSTM, and LSTM+CNN approaches. Here are additional research papers that focus on using deep learning techniques to detect several cyber attacks [29], [30], [31].

More specifically, the starting point of this work is represented by the Mobile Network operational dataset that Van Do et al. [32] have used to propose an ML-based detection system that employs a dedicated anomaly detector for each of the three primary contextual attributes indicating the presence of IMSI catchers. The above-mentioned paper's objective was to show the potential of applying machine learning techniques to facilitate the detection process and not complete implementation.

Inspired by the previous work mentioned above, we use the same dataset to study how temporal and convolutional deep learning classification models can be adopted and used for detecting suspicious (criminal) behaviour modelled as a multivariate time series. These models serve the purpose of assisting LEAs in identifying suspicious events indicative of IMSI catcher activity within a mobile network, thereby enabling the provision of timely preemptive alerts.

III. Methodology

Leveraging AI technologies such as ML, DL, Natural Language Processing, and behavioural analysis has transformed the cybersecurity landscape, offering a high level of adaptability, speed, and precision that was previously unattainable through conventional methods. This synergy between AI and cybersecurity encouraged us to aid LEAs and law enforcement officers (LEOs) with new innovative AI solutions in order to have the ability to anticipate, detect, and respond to criminal threats with expertise.

Building on this progress, we were inspired to apply DL techniques for time series data analysis like CNN and LSTM to broader applications. By combining CNN and LSTM for time series classification, we harness the strengths of both algorithms to create intelligent and effective solutions. The main goal of this work is not limited to IMSI catcher detection but extends to let CNN and LSTM be the foundation for a wider range of applications that can provide LEAs with powerful AI tools to process and analyze complex historical data, enabling them to identify suspicious deviations and enhance their investigative practices. This, in turn, contributes to preventing unlawful activities, supporting counterterrorism efforts, and enhancing public safety.

To our knowledge, the proposed LSTM and CNN-LSTM approaches are the first DL models of their kind, taking into account the temporal aspects of suspicious events for detecting IMSI catchers. To accomplish this mission, we have developed a methodological process as shown in Figure 1 comprising three main steps:

- Data Collection and Preprocessing: This step involves data cleaning, data transformation, data split and feature selection.
- Models' Configurations: In this step, we determine the architectures and model hyperparameters to match the experiment requirements.
- Model Evaluation: The model's performance is evaluated using relevant evaluation metrics.

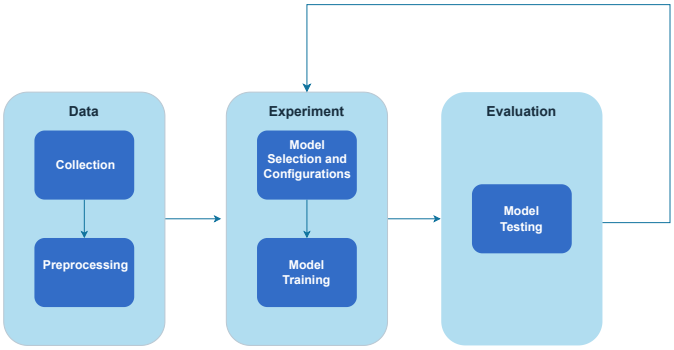

Fig. 1. Detailed flow of the overall methodological process

IV. Experimental Framework

In this section, we provide a detailed overview of the dataset collection in Subsection IV-A and present the carried data preprocessing in Subsection IV-B. The models' configurations are shown in Subsection IV-C, evaluation metrics are explained in Subsection IV-D, and in Subsection IV-E, we describe the two experiments conducted to evaluate the performance of the developed models:

A. Dataset

The dataset we are using was collected by reporters from a Norwegian newspaper called Aftenposten [2] during an investigation aimed at determining whether IMSI catchers were in use in the Norwegian capital, Oslo. Johansen et al. [33] used a GSMK CryptoPhone® Baseband Firewall (BBFW) [34], a device designed to protect against monitoring, eavesdropping, and tapping of mobile communications, while also alerting the user to potential IMSI catcher attacks. As previously discussed in Subsection II-A, the CryptoPhone checks whether signals originate from approved mobile operator base stations by detecting various attributes that indicate the presence of an IMSI catcher, including signal strength and ID number matching. It also detects if mobile network encryption is turned off or if the mobile device is transitioning from 3G/4G to 2G, which could be a sign of eavesdropping. Additionally, it identifies if the mobile device is communicating on the baseband and sending data without being in active use. The log data from the CryptoPhone was cleaned, preprocessed, and parsed to extract discriminative features and structure it into an ML-usable format. The dataset has several primary features that form the basis of the classification process, including the previously mentioned attributes that indicate the presence of an IMSI catcher. These features also include the activity time of baseband processor in seconds, inbound and outbound phone or data activity, signal strength, network operator, and the source of information, whether it comes from the baseband's operating system or the mobile's operating system. The general classification of events according to the Technical Briefing of GSMK CryptoPhone Baseband Firewall [35] is as follows:

- Class *None* represents the normal events.
- Class *Low* represents events with very low probability to be suspicious.
- Class *Medium* represents events with medium probability to be suspicious.
- Class *High* represents events with high probability to be suspicious.
- Class *Very High* represents suspicious events.

B. Data Preprocessing

Fortunately, the dataset is well-organized, requiring minimal cleaning since the original text data extracted from Cryptophone has already been transformed into categorical values. Firstly, we explore the dataset to understand its structure and the types of features present. All important features have categorical values except two, which represent the activity time in seconds and signal strength in ASU (Arbitrary Strength Unit). Some features were cleaned by removing all irrelevant text data (Nan) and imputing missing values with zeros so only the numerical data are kept to have a reliable analysis. The dataset's class distribution reveals a significant imbalance issue where class *none* vastly outnumber the other four classes. In our case, we are primarily interested in identifying the events that need immediate attention due to their high suspicion levels and highlight them for further investigation by LEAs, rather than categorizing them into multiple subgroups based on severity of threat. Thus, the problem is framed as a binary classification which is more practical for our real-world scenario. The class *none* accounts for 97.5% of the data instances, with the remaining 2.5% distributed among the others. 70% of the dataset is allocated for training, and the remaining 30% is reserved for testing the model, with 5% of the training data designated for validation.

C. Configuration

In this subsection, we present an overview of nine different architectures that will be evaluated and compared for their performance in time series classification. As shown in Table I, the first model, CNN I, is a basic CNN network which consists of a 1D convolutional input layer, and a dense output layer for making binary classification. Model CNN II includes an additional dense hidden layer with 128 neurons. LSTM I consists of only an LSTM input layer and a dense output layer, whereas LSTM II introduces an additional hidden dense layer with 128 neurons. Moving on to the fifth through eighth model architectures, we leverage the potentialities of CNN+LSTM or LSTM+CNN cascade hybrid architecture by using the output of the CNN or LSTM layer as the input for the subsequent

[2]https://www.aftenposten.no/

LSTM or CNN layer, respectively. CNN+LSTM I has a 1D CNN input layer, LSTM hidden layer and dense output layer, whereas CNN+LSTM II has a 1D CNN input layer, LSTM and Dense as hidden layers, and again dense as the output layer. LSTM+CNN I has an LSTM input layer, 1D CNN hidden layer with 128 neurons, and dense output layer, while LSTM+CNN II has an LSTM input layer, 1D CNN and dense with 128 neurons as hidden layers, and dense output layer. The CNN layer is responsible for extracting local features and patterns, while the LSTM layer captures global and sequential dependencies. In that sense, we can verify the importance of the order of CNN and LSTM layers in effecting classification results.

Finally, we construct a parallel architecture consisting of two CNN and LSTM branches, each with its set of layer(s) to process the input data independently. These branches are concatenated before the final output layer for classification. All 1D CNN layers have 128 filters, LSTM layers have 128 neurons and lastly, all dense output layers have one neuron that outputs the classification probability between 0 and 1, outputs with a probability higher than 0.5 are classified as suspicious.

TABLE I
OVERVIEW OF MODELS ARCHITECTURES

Architecture	Input Layers	Hidden Layers (neurons)	Output Layers
CNN I	1D CNN	-	Dense
CNN II	1D CNN	Dense (128)	Dense
LSTM I	LSTM	-	Dense
LSTM II	LSTM	Dense (128)	Dense
CNN+LSTM I	1D CNN	LSTM	Dense
CNN+LSTM II	1D CNN	LSTM+Dense (128)	Dense
LSTM+CNN I	LSTM	1D CNN	Dense
LSTM+CNN II	LSTM	1D CNN+ Dense (128)	Dense
Parallel CNN/LSTM	CNN LSTM	Dense (128)	Dense

D. Evaluation Metrics

The evaluation process of our models is mainly based on the confusion matrix which provides a tabular representation that summarizes the model's predictions and actual class labels. The rows of the matrix represent the number of predicted classes, while the columns represent the actual class.

From the confusion matrix values, we can extract some useful metrics that can be used to effectively evaluate the performance of the developed models. Here are some commonly used metrics [36]:

- *Precision*: it calculates the ratio of correctly classified positives over total positive predictions.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

- *Recall*: it calculates the ratio of correctly classified positives over total actual positives.

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

- *F1-Score*: it provides a balanced measure by considering the harmonic mean of precision and recall.

$$F1 - Score = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{3}$$

- *Matthews Correlation Coefficient (MCC)*: it assesses the quality of binary classification, taking into account both false positives and false negatives. This makes it particularly useful for evaluating model performance in situations where the class distribution is imbalanced. MCC score ranges from -1 to 1: 1 means a perfect prediction, 0 means the prediction is random, and -1 means the model does the opposite of what it should.

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{4}$$

The precision, recall, and F1-score percentages reported in the paper represent the macro average scores for the two classes.

E. Experiments

1) ***Class Ambiguity****:* The first experiment focuses on three case studies aimed at accurately assigning class labels to each data sample. Classes *low* and *medium* do not have clear assignments to the suspicious or non-suspicious categories, leading to uncertainty in labeling these data samples. This ambiguity revolves around whether data samples belonging to these classes should be classified as non-suspicious or suspicious. To address this issue, we need to refine the class definitions across three case studies as illustrated in Figure 2 and observe how the developed models perform. The case study yielding the best results will surely provide us with the correct label assignment, and the top two models from this case study will be selected for the second experiment. In the first case study (CS I), only class *none* is categorized as non-suspicious, while the remaining four classes are labeled as suspicious. In the second case study (CS II), class *low* is grouped with class *none* in the non-suspicious category, while *medium*, *high*, and *very high* classes are assigned to the suspicious category. Lastly, in the third case study (CS III), only *high* and *very high* classes are considered suspicious, while the others are assigned to the non-suspicious category. As the introduced architectures in Subsection IV-C. For each configuration, the experiment is repeated three times to assess the repeatability of the results and the randomness of the solutions.

2) ***Temporal Length****:* The objective of this second experiment is to train the two models that exhibited the best performance in terms of precision and recall simultaneously (F1-Score) in the previous experiment with five different sequence lengths (3, 5, 10, 20, 25) to find the optimal temporal length of each model for our specific task. Given the significance of the temporal order of data samples, and in order to capture the temporal patterns and relations in the time series data, it is

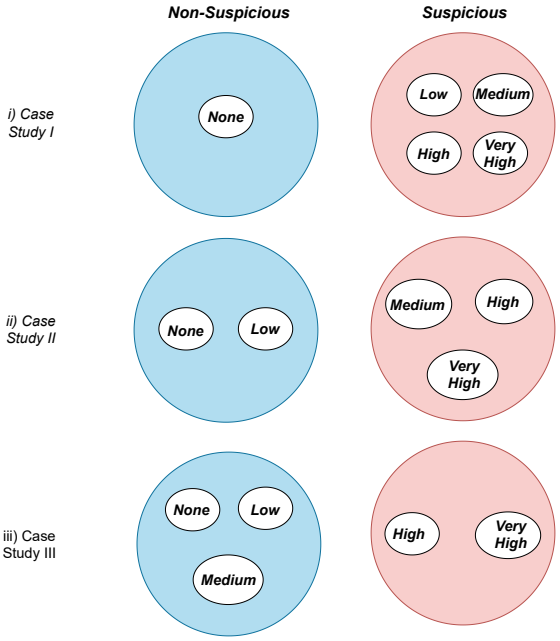

Fig. 2. Visual illustration of classes definitions refinement in the first experiment

essential that the input fed into the model should be in the form of a sequence. This sequence represents the history length or time steps the model considers when making predictions. Consequently, we use a sliding window to create overlapping sequences of the dataset with adjustable length that moves step-by-step through the entire series. Additionally, we conducted three tests for each architecture at every sequence length to ensure result repeatability.

V. Experimental Study

In this section, we present the evaluation results of the class ambiguity experiment in Subsection V-A and the temporal length experiment in Subsection V-B:

A. *Class Ambiguity*

As we can see from the results of CS I in Table II, LSTM+CNN II and Parallel CNN/LSTM are the top-performing architectures with the highest F1-Score (83.67%) and the highest MCC (0.68) among all configurations followed by LSTM II with f1-Score equals (83%). This suggests that combining LSTM and CNN or using parallel CNN/LSTM layers yields the best overall performance for the classification task in Case Study I.

As depicted in Table III, LSTM II stands out as the top-performing architecture in Case Study II. It achieves a precision of 87%, indicating a low rate of false positives (suspicious instances that are incorrectly predicted), and a recall of 87.33%, indicating it captures a significant portion of suspicious instances. The F1-Score is 86.67%, reflecting a well-balanced performance between precision and recall. The MCC is the highest among all architectures at 0.74, indicating a very strong agreement between predictions and actual outcomes. followed by the hybrid architecture CNN+LSTM I with a slight difference.

TABLE II
Overview of weighted average metrics for CS I

Case Study I				
Architecture	Precision	Recall	F1-Score	MCC
CNN I	83.67	74.33	77.33	0.57
CNN II	87.33	73.33	78.67	0.59
LSTM I	87.33	79.33	82.67	0.66
LSTM II	86.33	80.67	83	0.67
CNN+LSTM I	88.33	73.33	79	0.60
CNN+LSTM II	**90.33**	74.33	79.67	0.62
LSTM+CNN I	88	76.33	80.67	0.63
LSTM+CNN II	**88.33**	**80**	**83.67**	**0.68**
Parallel CNN/LSTM	87.67	**80.67**	**83.67**	**0.68**

TABLE III
Overview of weighted average metrics for CS II

Case Study II				
Architecture	Precision	Recall	F1-Score	MCC
CNN I	79.6	84.6	81.33	0.63
CNN II	86	81.33	83.33	0.67
LSTM I	86.67	86.33	85.67	0.72
LSTM II	**87**	**87.33**	**86.67**	**0.74**
CNN+LSTM I	**86.33**	**86.67**	**86.67**	**0.73**
CNN+LSTM II	**90.67**	79.33	83.67	0.69
LSTM+CNN I	84.67	86	85.33	0.70
LSTM+CNN II	85	84.67	84.67	0.70
Parallel CNN/LSTM	86.67	87	86.33	0.73

TABLE IV
Overview of weighted average metrics for CS III

Case Study III				
Architecture	Precision	Recall	F1-Score	MCC
CNN I	90	81.67	84.67	0.70
CNN II	94.33	83	87.67	0.76
LSTM I	90	87	88.33	0.77
LSTM II	**97**	**85.33**	**90.33**	**0.81**
CNN+LSTM I	86	83	84	0.69
CNN+LSTM II	95.67	82.67	88	0.77
LSTM+CNN I	92.67	85.3	88.33	0.77
LSTM+CNN II	93	85.67	88.67	0.78
Parallel CNN/LSTM	94.67	85	89	0.79

LSTM II emerges as the leading architecture in CS III as shown in Table IV. It attains a precision rate of 97%, signifying an exceptionally low occurrence of false positives, alongside a recall rate of 85.33%. The resulting F1 Score, at 90.33%, indicates a balanced performance that encompasses both precision and recall. Parallel CNN/LSTM architecture shows strong performance with a precision of 94.67% and a recall of 85%. The F1-Score is 89%, indicating a balanced performance. The MCC is 0.79, emphasizing that combining CNN and LSTM improves the overall performance of the model.

B. *Temporal Length*

In this experiment, we trained the two top-performing models of CS III, LSTM II and Parallel CNN/LSTM, using five different sequence lengths (3, 5, 10, 20, and 25) to assess

how this parameter can impact classification quality. Figure 3 shows how the precision of two models, LSTM II and Parallel CNN/LSTM changes as the sequence length varies. For most sequence lengths, LSTM II got better precision except for 25 we observed a steep drop compared to Parallel CNN/LSTM.

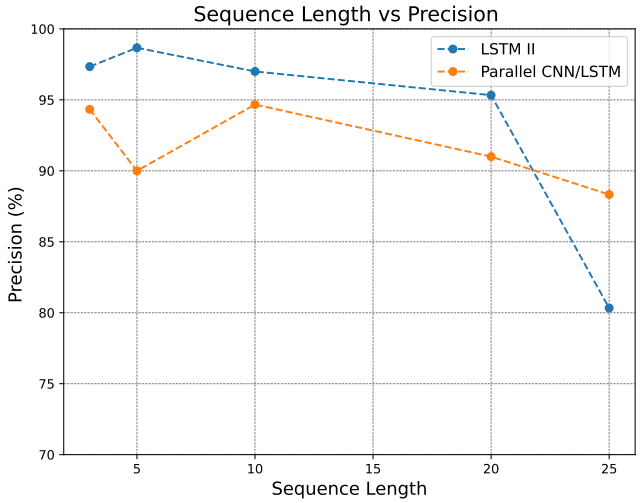

Fig. 3. Impact of Sequence Length on Precision: LSTM II vs. Parallel CNN/LSTM

Figure 4 illustrates recall values as the sequence length varies for the two models. For LSTM II, recall remains relatively stable as sequence length increases from 3 to 20. However, there is a noticeable drop in recall when the sequence length is 25. while on the other hand, Parallel CNN/LSTM, shows some variation in recall percentages with changing sequence length. It starts at a relatively high value and then gradually decreases as sequence length increases. Overall, "LSTM II" tends to have higher recall compared to Parallel CNN/LSTM at specific lengths (10 and 20), while Parallel CNN/LSTM performs better when the sequence length reaches 25.

Across most sequence lengths, the LSTM II model tends to have a slightly higher F1-Score compared to the Parallel CNN/LSTM model as shown in Figure 5. Notably, both models experience a significant drop in F1-Score when the sequence length is 25, indicating that longer sequence lengths may not be optimal for this task.

We believe that the Parallel CNN/LSTM model performs better than LSTM II at longer sequences due to the dimensionality reduction that CNN layers combined with pooling layers can do in the input data while retaining essential features. This reduction makes it more manageable for the LSTM part of the model to process longer sequences, as it deals with lower dimensional representations.

C. Discussion

The results affirm the transformative potential of AI in enhancing LEAs' capabilities and contribute to the broader

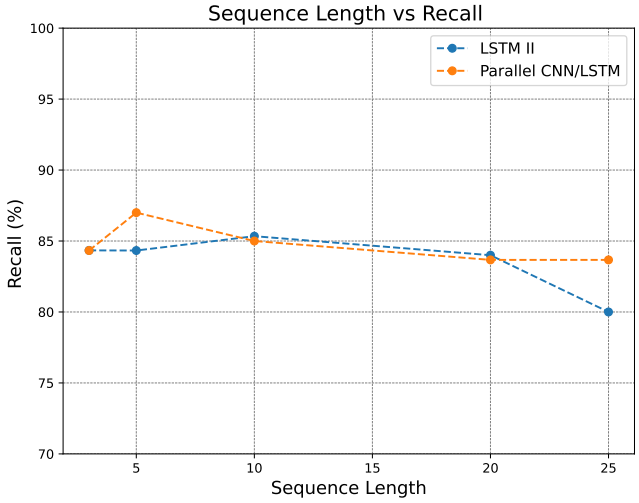

Fig. 4. Impact of Sequence Length on Recall: LSTM II vs. Parallel CNN/LSTM

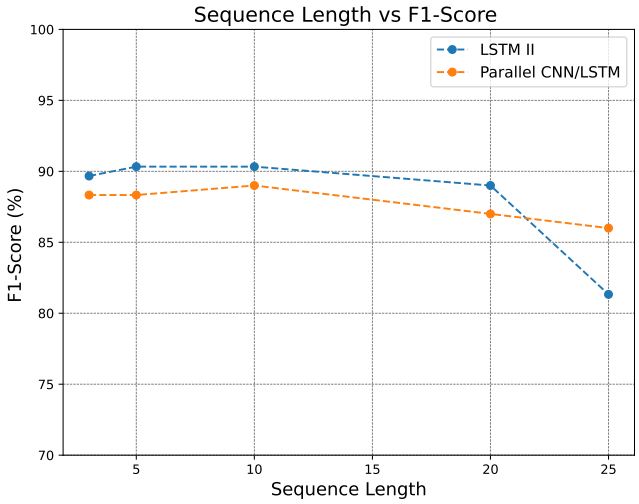

Fig. 5. Impact of Sequence Length on Precision: LSTM II vs. Parallel CNN/LSTM

mission of improving security and public safety. Furthermore, the presented models are intended to serve as the foundation for the practical deployment of a wide range of AI tools that include time related data analysis to enhance LEAs' autonomy in dealing with suspicious threats in public spaces. Notably, LEAs necessitate legal access to mobile networks to collect and feed data into the proposed models for the detection process of any suspicious behaviors within the network. Operational data collection of mobile networks by LEAs should follow legal regulations, ethical considerations and legitimate purposes while implementing these models. This entails obtaining legal authorization, ensuring privacy protection, minimizing data collection, and complying with

human rights standards to balance between public security needs and the protection of individual privacy rights. However, the initial experiment results reveal that CNN architectures exhibit lower performance when compared to hybrid CNN and LSTM architectures, as well as parallel CNN/LSTM models. This suggests that the addition of LSTM layers has notably improved the overall performance of the models. Additionally, the results highlight certain limitations of this research study. One limitation is related to the class overlap problem observed in the class medium. The classification results improved in CS III compared to CS I and CS II, where those data points were assigned to the non-suspicious class. To sum up, the high class exhibited a stronger association with the suspicious class compared to the medium class, despite both classes representing possible suspicious events but with less probability. The Second limitation is related to that many European countries are planning to gradually shut down 2G and 3G networks and transition to more advanced and secure generations like 4G and 5G which are better equipped to mitigate the threat of IMSI catcher attacks even though GSM is still the standard service in most parts of the world. This transition may impact the relevance of the research findings in regions where older network technologies are still in use. However, the results of the second experiment show that the temporal length is a double-edged sword parameter, it can affect the quality of classifications either positively or negatively. Longer sequences often lead to better predictions because they provide more information about the underlying patterns and relationships in the data but at the same time, they may increase the risk of overfitting and the model start memorizing the training data rather than learning general patterns. The results of the second experiment clearly demonstrate that, in our case, there is a drop in classification quality as the sequence length increases due to the imbalanced nature of the dataset. LSTM II outperformed Parallel CNN/LSTM across all proposed lengths except for 25.

VI. Conclusion

In this paper, we present a DL approach for IMSI catcher detection using CNN and LSTM algorithms. Our comparative study evaluates CNN, LSTM, hybrid cascade CNN+LSTM, and parallel CNN/LSTM models for multivariate time series classification. Notably, our study highlights the significant potential of LSTM II and Parallel CNN/LSTM models, achieving F1-Score percentages of 90.33% and 89%, respectively, in detecting highly suspicious events that may signal IMSI catcher presence. Furthermore, the paper also reveals the negative impact on LSTM II and Parallel CNN/LSTM models' performance if we increase the sequence length. This work is the first DL IMSI catcher detection network-based solution that takes into account the temporal patterns of suspicious events that may be timely associated with other attacks instead of conventional portable devices monitoring radio access networks. These models go beyond IMSI catcher detection and play a crucial role in enhancing security, preventing crime, supporting counterterrorism, generating accurate predictions, mitigating human bias in decision-making, and ultimately making the public space safer. By leveraging the proposed CNN and LSTM models, LEAs can gain access to advanced analytical tools capable of processing, analyzing, and interpreting complex historical time series data sources that empower them to identify suspicious deviations effectively and enhance their investigative practices. However, future research directions should explore novel data preprocessing approaches to address the class overlap problem and adopt new resampling techniques for categorical time series data to prevent the creation of a non-representative dataset that could lead to the loss of its real-world temporal patterns, ensuring more robust and accurate analysis. Additionally, we need to develop more advanced AI models to detect and prevent evolving security threats in advanced mobile technologies such as 5G. The limited access to data for AI technologies in crime and counterterrorism applications is primarily due to data sensitivity and privacy concerns. Such data often contains personal information, surveillance data, or criminal records. Balancing security needs with data access remains a complex challenge. To advance the effective use of predictive AI in LEA operations, concerted efforts and dedicated research are needed to tackle these complex issues and bridge the gap between data access and security to expand the AI applicability to address various evolving threats in LEAs practices.

Acknowledgment

This work has been partially supported by the European Union's Horizon 2020 Research and Innovation Program under the project STARLIGHT (Grant Agreement No. 101021797)

References

[1] M. N.-E. Saulaiman, M. Kozlovszky, and Á. Csilling, "A survey on vulnerabilities and classification of cyber-attacks on 5g-v2x," in *2021 IEEE 21st International Symposium on Computational Intelligence and Informatics (CINTI)*. IEEE, 2021, pp. 000 235–000 240.

[2] B. Hamid, N. Jhanjhi, M. Humayun, A. Khan, and A. Alsayat, "Cyber security issues and challenges for smart cities: A survey," in *2019 13th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics (MACS)*. IEEE, 2019, pp. 1–7.

[3] S. Pazouki, K. Bibek, H. A. Alkhwaildi, and A. Asrari, "Modelling of smart homes affected by cyberattacks," in *2020 52nd North American Power Symposium (NAPS)*. IEEE, 2021, pp. 1–6.

[4] Y. C. Tok and S. Chattopadhyay, "Identifying threats, cybercrime and digital forensic opportunities in smart city infrastructure via threat modeling," *Forensic Science International: Digital Investigation*, vol. 45, p. 301540, 2023.

[5] S. Raaijmakers, "Artificial intelligence for law enforcement: challenges and opportunities," *IEEE security & privacy*, vol. 17, no. 5, pp. 74–77, 2019.

[6] M. Boukabous and M. Azizi, "Image and video-based crime prediction using object detection and deep learning," *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 3, pp. 1630–1638, 2023.

[7] A. S. Irwin and A. B. Turner, "Illicit bitcoin transactions: challenges in getting to the who, what, when and where," *Journal of money laundering control*, vol. 21, no. 3, pp. 297–313, 2018.

[8] J. J. Bryson and A. Theodorou, "How society can maintain human-centric artificial intelligence," *Human-centered digitalization and services*, pp. 305–323, 2019.

[9] Kaspersky, "Top seven mobile security threats," 2022.

[10] S. Park, A. Shaik, R. Borgaonkar, and J.-P. Seifert, "Anatomy of commercial imsi catchers and detectors," in *Proceedings of the 18th ACM Workshop on Privacy in the Electronic Society*, 2019, pp. 74–86.

[11] H. Alrashede and R. A. Shaikh, "Imsi catcher detection method for cellular networks," in *2019 2nd International Conference on Computer Applications & Information Security (ICCAIS)*. IEEE, 2019, pp. 1–6.

[12] X. Jin, X. Yu, X. Wang, Y. Bai, T. Su, and J. Kong, "Prediction for time series with cnn and lstm," in *Proceedings of the 11th international conference on modelling, identification and control (ICMIC2019)*. Springer, 2020, pp. 631–641.

[13] B. Zhao, H. Lu, S. Chen, J. Liu, and D. Wu, "Convolutional neural networks for time series classification," *Journal of Systems Engineering and Electronics*, vol. 28, no. 1, pp. 162–169, 2017.

[14] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[15] T. Rademacher, "Artificial intelligence and law enforcement," *Regulating artificial intelligence*, pp. 225–254, 2020.

[16] L. Elluri, V. Mandalapu, and N. Roy, "Developing machine learning based predictive models for smart policing," in *2019 IEEE International Conference on Smart Computing (SMARTCOMP)*. IEEE, 2019, pp. 198–204.

[17] S. A. Chun, V. Avinash Paturu, S. Yuan, R. Pathak, V. Atluri, and N. R. Adam, "Crime prediction model using deep neural networks," in *Proceedings of the 20th Annual International Conference on digital government research*, 2019, pp. 512–514.

[18] M. Saraiva, I. Matijošaitienė, S. Mishra, and A. Amante, "Crime prediction and monitoring in porto, portugal, using machine learning, spatial and text analytics," *ISPRS International Journal of Geo-Information*, vol. 11, no. 7, p. 400, 2022.

[19] W. A. Al-Khater, S. Al-Maadeed, A. A. Ahmed, A. S. Sadiq, and M. K. Khan, "Comprehensive review of cybercrime detection techniques," *IEEE Access*, vol. 8, pp. 137 293–137 311, 2020.

[20] C.-S. Sung and J. Y. Park, "Design of an intelligent video surveillance system for crime prevention: applying deep learning technology," *Multimedia Tools and Applications*, pp. 1–13, 2021.

[21] K. K. Kumar and H. Venkateswara Reddy, "Crime activities prediction system in video surveillance by an optimized deep learning framework," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 11, p. e6852, 2022.

[22] S. T. Ratnaparkhi, A. Tandasi, and S. Saraswat, "Face detection and recognition for criminal identification system," in *2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*. IEEE, 2021, pp. 773–777.

[23] S. Sandhya, A. Balasundaram, and A. Shaik, "Deep learning based face detection and identification of criminal suspects." *Computers, Materials & Continua*, vol. 74, no. 2, 2023.

[24] F. Zola, L. Segurola-Gil, J. L. Bruse, M. Galar, and R. Orduna-Urrutia, "Attacking bitcoin anonymity: generative adversarial networks for improving bitcoin entity classification," *Applied Intelligence*, vol. 52, no. 15, pp. 17 289–17 314, 2022.

[25] S. Tobiyama, Y. Yamaguchi, H. Shimada, T. Ikuse, and T. Yagi, "Malware detection with deep neural network using process behavior," in *2016 IEEE 40th annual computer software and applications conference (COMPSAC)*, vol. 2. IEEE, 2016, pp. 577–582.

[26] B. Kolosnjaji, A. Zarras, G. Webster, and C. Eckert, "Deep learning for classification of malware system call sequences," in *AI 2016: Advances in Artificial Intelligence: 29th Australasian Joint Conference, Hobart, TAS, Australia, December 5-8, 2016, Proceedings 29*. Springer, 2016, pp. 137–149.

[27] P. Sun, P. Liu, Q. Li, C. Liu, X. Lu, R. Hao, and J. Chen, "Dl-ids: Extracting features using cnn-lstm hybrid network for intrusion detection system," *Security and communication networks*, vol. 2020, pp. 1–11, 2020.

[28] Z. Alshingiti, R. Alaqel, J. Al-Muhtadi, Q. E. U. Haq, K. Saleem, and M. H. Faheem, "A deep learning-based phishing detection system using cnn, lstm, and lstm-cnn," *Electronics*, vol. 12, no. 1, p. 232, 2023.

[29] M. Saed and A. Aljuhani, "Detection of man in the middle attack using machine learning," in *2022 2nd International Conference on Computing and Information Technology (ICCIT)*. IEEE, 2022, pp. 388–393.

[30] R. Pascanu, J. W. Stokes, H. Sanossian, M. Marinescu, and A. Thomas, "Malware classification with recurrent networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 1916–1920.

[31] C. Gupta, I. Johri, K. Srinivasan, Y.-C. Hu, S. M. Qaisar, and K.-Y. Huang, "A systematic review on machine learning and deep learning models for electronic information security in mobile networks," *Sensors*, vol. 22, no. 5, p. 2017, 2022.

[32] T. Van Do, H. T. Nguyen, N. Momchil, and V. T. Do, "Detecting imsi-catcher using soft computing," in *Soft Computing in Data Science: First International Conference, SCDS 2015, Putrajaya, Malaysia, September 2-3, 2015, Proceedings 1*. Springer, 2015, pp. 129–140.

[33] P. A. Johansen, A. B. Foss, and F. H. Thoresen, "Aftenposten data set," https://mm.aftenposten.no/mobilspionasje/, accessed: July 10, 2023.

[34] "Gsmk cryptophone," https://www.cryptophone.de/.

[35] "Gsmk cryptophone," https://telephone-museum.org/wp-content/uploads/2014/12/GSMK-Baseband-Firewall-Technical-Briefing.pdf.

[36] M. Hossin and M. N. Sulaiman, "A review on evaluation metrics for data classification evaluations," *International journal of data mining & knowledge management process*, vol. 5, no. 2, p. 1, 2015.