



Deep Learning-Based DDoS-Attack Detection for Cyber–Physical System Over 5G Network

Bilal Hussain , *Student Member, IEEE*, Qinghe Du , *Member, IEEE*,
Bo Sun, *Member, IEEE*, and Zhiqiang Han

Abstract—With the advent of 5G, cyber–physical systems (CPSs) employed in the vertical industries and critical infrastructures will depend on the cellular network more than ever; making their attack surface wider. Hence, guarding the network against cyberattacks is critical not only for its primary subscribers but to prevent it from being exploited as a proxy to attack CPSs. In this article, we propose a consolidated framework, by utilizing deep convolutional neural networks (CNNs) and real network data, to provide early detection for distributed denial-of-service (DDoS) attacks orchestrated by a botnet that controls malicious devices. These puppet devices individually perform silent call, signaling, SMS spamming, or a blend of these attacks targeting call, Internet, SMS, or a blend of these services, respectively, to cause a collective DDoS attack in a cell that can disrupt CPSs' operations. Our results demonstrate that our framework can achieve higher than 91% normal and underattack cell detection accuracy.

Index Terms—5G, artificial intelligence (AI), call detail record (CDR), convolutional neural networks (CNNs), cyber–physical system (CPS), cybersecurity, DDoS attack, deep learning (DL).

I. INTRODUCTION

CYBER–PHYSICAL system (CPS) is a complex, large, and networked amalgam of sensors, actuators, and computing nodes that monitor and control physical processes [1], [2]. Because of its highly intricate and heterogeneous nature, contributed by both cyber and physical aspects, a CPS has many general and application-specific vulnerabilities that can be exploited by an attacker to perform mischievous acts [1, Sec. IV and V]. Since CPSs control physical processes, the consequence

of an attack can be irreversible and disastrous depending on the severeness of the attack and application domain: industrial control system (ICS), smart city, intelligent transportation, etc. CPS innovations applied in vertical industries/sectors (energy, automotive, eHealth, manufacturing, etc.) will potentially account for more than \$82 trillion in economic activity by 2025 [3]; sabotaging a CPS equates to a significant bump on an economy and, hence, its security must be of the paramount importance.

5G is currently being devised to be a disruptive technology that will play a crucial role in providing communications for CPSs and to enable new services in the applied verticals (see [4, Fig. 4] for an integrated 5G architecture). Consequently, this will increase their dependence on cellular networks [5], which can also be inferred from use cases pertaining to the various verticals described in 5G infrastructure public private partnership (5G PPP) white papers [6]. 5G is anticipated to meet the stringent requirements of industries: ultrareliability, low latency, high density of connected devices, etc., [4], [7]. In fact, 5G will provide services for real-time and mission-critical applications: state estimation in smart grids [8], assisted overtaking in smart vehicles [5], etc.

Various attacks [9], [10, Table 1] can be launched against cellular networks that compromise availability, integrity, or confidentiality. Denial-of-Service (DoS) attack targets the availability of network resources in a region and has potential to raze a network: in case when there are multiple such attacks in dispersed and well-synchronized manner called distributed denial-of-service (DDoS) attack [9], [10]. According to a report by Verizon [11], DDoS attacks topped the list of most frequent cybersecurity incidents of 2017. They can be devised as beachhead or smoke screen for IT security experts, with which some other objective(s) (for example, data breach) can be accomplished [11], [12]: in cellular networks, the DDoS attack cannot just heavily affect the network and its legitimate users but can have potential side effects in disrupting CPSs that heavily rely on the networks. Once compromised, cellular networks can be exploited as powerful attack vectors against CPSs; hence, strong measures should be taken such that they cannot be exploited as proxy to attack CPSs.

Apart from a zero-day vulnerability, malicious user(s) can also exploit known vulnerabilities in the cellular network to orchestrate DDoS attack—its mitigation in 4G network is yet an open issue [9]. Diverse individual attacks such as silent call [13], signaling [14], and SMS flooding [15] attacks (elaborated in Section I-A) can be staged by utilizing a network of bots known

Manuscript received June 27, 2019; revised September 24, 2019 and January 13, 2020; accepted February 2, 2020. Date of publication February 17, 2020; date of current version November 18, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61941119, in part by the ZTE Industry-Academic-Research Cooperation Funds, and in part by the Fundamental Research Funds for the Central Universities, China. Paper no. TII-19-2742. (*Corresponding author: Qinghe Du.*)

Bilal Hussain and Qinghe Du are with the School of Information and Communications Engineering, Xi'an Jiaotong University, Xi'an 710000, China, and also with the Shaanxi Smart Networks and Ubiquitous Access Research Center, Xi'an 710049, China (e-mail: bilalhussain@stu.xjtu.edu.cn; duqinghe@mail.xjtu.edu.cn).

Bo Sun and Zhiqiang Han are with the ZTE Corporation, Shenzhen 518057, China (e-mail: sun.bo1@zte.com.cn; han.zhiqiang1@zte.com.cn).

Color versions of one or more of the figures in this article are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TII.2020.2974520

as botnet [an overlay network comprising large number of malware-infected mobile devices that can receive commands from a botmaster (cybercriminal)] to participate in a collective DDoS attack [9], [10], [16]. These devices can be infected by utilizing SMS, email attachments, or other means to spread and inject malware [10] and this could be accomplished in different environments such as in device-to-device (D2D) networks [17], etc. The severity of the threat from botnets can be realized in [11, Fig. 17], which elucidates global botnet breaches in 2017.

Artificial intelligence (AI) is a superset of machine learning, itself a superset of more powerful algorithms and techniques known as deep learning (DL) [18]. There is a recent surge of interest and clamor, among researchers from both academia [19] and industry/nation state [20], in utilizing AI/DL for the protection of CPSs—especially the ones utilized in the critical infrastructures such as smart grids and financial networks—against cyberattacks. Proliferating success of DL is not just visible in image recognition field [18] but in all walks of life, evident from the increased number of AI patent applications granted worldwide, shown in [21, Fig. 1].

Motivated by the abovementioned reasons, a novel DL [in particular, convolutional neural network (CNN)] based framework for the detection of the silent call, signaling, and SMS flooding attacks that can collectively or individually cause a DDoS attack against 4G network infrastructure, disrupting services not only for the subscribers but also for the cellular-dependent CPS devices is proposed by us. As compared to many computationally expensive content-based schemes that also consequently compromise user privacy [22], [23], our solution is lightweight because: 1) it is independent of any individual activity (call, SMS, or Internet) content; and 2) it leverages already-available call detail record (CDR) data in the cellular network (mainly utilized for customer billing purpose), instead of depending on dataset that demands additional resources (observation time, computation, communication, etc.) for its acquisition. CDRs contain measure of subscribers' interactions with the network in a spatiotemporal manner that can infer normal cell's behavior, and can be utilized to identify an underattack cell's behavior [22], [24].

This article makes the following prominent contributions:

- 1) proposes a novel and consolidated framework for the detection of silent call, signaling, SMS flooding, and a blend of these attacks that ultimately cause a DDoS attack across the cellular infrastructure;
- 2) presents a scalable and expandable solution for the attacks detection by utilizing CNN, for which the input images can be expanded to include a greater number of cells without modifying the model;
- 3) deploys a state-of-the-art very deep CNN model called residual network with 50 layers (ResNet-50) and also introduces a relatively simple model called deep rudimentary CNN (DRC) model having six layers that yields better detection accuracy for most of the attack scenarios.

Next, we describe the attacks we are dealing with throughout this article in the remaining portion of this section. Then, we summarize the relevant work in Section II followed by a discussion on preliminaries to our proposed method in Section III. Then, we explain our framework's implementation

in Section IV. We subsequently elaborate the results and our framework's performance evaluation in Section V. Finally, we discuss our results, future insights, and draw the concluding remarks in Section VI.

A. Description of the Attacks

1) *Silent Call Attack*: This attack is launched by exploiting a fundamental design flaw in voice over LTE (VoLTE, a voice solution proposed for 4G LTE network) call establishment procedure [13, Fig. 4]. During the procedure, initiated after a VoLTE-supported device calls its victim, a number of messages are exchanged involving caller, callee, VoLTE server, and gateways. In between, the network reserves resources for the call before caller device sends a "session initiation protocol (SIP) update" message that eventually enables the callee's device to ring—at this instance, the caller avoids sending the message to bypass the ringing. As a consequence and since the network had already reserved the resources to carry out the call, the victim's device is compelled to be stuck in a high-powered radio resource control (RRC) state without the callee's knowledge [13]. An Android application called *VoLTECaller* has also been developed for the demonstration of this attack [25].

2) *Signaling Attack*: This attack targets to overload a core network (CN) element (e.g., a gateway) that processes RRC-based signaling messages—exchanged among different network entities for the purpose of efficient resource management [10]. During the attack, a malicious user requests for a bearer setup (known as random access) to send data and have "connected" state; after the (resource) allocation, it just waits until the timeout and continuously repeats this process. This phenomenon generates huge signaling messages for the network entities to process: a total of 24 messages are required for a bearer activation and deactivation [26]. Since an LTE/LTE-A user can initiate multiple bearers (max. eight), this amplifies the number of generated messages.

3) *SMS Flooding Attack/SMS Spamming [Toward IP Multimedia Subsystem (IMS)] Attack*: This attack relies on security vulnerabilities stemmed from technology migration: from the circuit-switched (CS) based network (3G) carrying SMS via control-plane to IMS and the packet-switched-based network (4G) carrying SMS via data-plane. In this attack, huge amount of forged SIP/SMS messages are injected during an SIP session (initiated during a SMS exchange) between the device's SMS client and IMS server [10], [27]; aiming to computationally overload the IMS server.

II. RELEVANT WORK

In the literature, detections of the silent call, signaling, and SMS flooding attacks have been mostly considered individually and to our knowledge, no consolidated framework has been proposed for their detection. Therefore, we study each of them separately. For their detection, many studies in the past have utilized content-based approaches in which the actual contents of the user activities (IP packets, SMS messages, etc.) are analyzed [22], [23]. However, such techniques have high computation cost and might be infeasible in practical settings.

The severity of silent call attack in 4G LTE networks has been thoroughly discussed by Tu *et al.* [13], and in their recent extended work [25]. Ruan *et al.* [23] utilized game theory to detect silent call attack by monitoring peak value and variation trend of traffic data volume. They claimed to have a lightweight solution in contrast to the past studies that proposed computationally exhaustive content-based solutions.

For signaling attacks, Bang *et al.* [28] proposed a detection scheme based on a hidden semi-Markov model by utilizing bearer wakeup packet generation rate in the wireless sensor and actuator network (WSAN). The scheme requires training examples constructed from the historical network data, which may take several hours or days of observation to acquire. Bassil *et al.* [26] utilized number of bearer requests per user per minute as a criterion to determine signaling attack: attack is detected if the amount exceeds a threshold. However, the information on how to determine the threshold is missing; it is important as it will directly affect the detector's performance. Gupta *et al.* [14] analyzed subset of a user device's IP packets using the support vector machine based method to detect signaling DDoS attacks. However, the proposed method might be computationally expensive due to its content-based detection nature.

SMS spamming (toward IMS) attack mechanism is discussed exhaustively in [27] and its detection is studied by Papadopoulos *et al.* [22], [24]. In [22], they simulated SMS flooding and also signaling attacks, and generated the synthetic CDR data accordingly—having the effects of attacks reflected on the relevant user activities. They proposed a graph-based descriptor to identify the anomalous mobile devices within 1-h slot in the data pertaining to a single cell. In their extended work [24], they identified a group of SMS flooding attackers by clustering users into different groups, each representing a unique traffic behavior. Similarly, Murynets *et al.* [15] utilized a graph-based clustering method and SMS CDR data to detect SMS flooding attack that caused DDoS in the machine-to-machine (M2M) communications network.

Our research is distinguished from all the abovementioned studies: it utilizes CDR dataset (lighter solution by avoiding content-based techniques) to detect SMS flooding attack in addition to silent call and signaling attacks (unified framework) within a 10-min slot (faster detection) in multiple cells simultaneously (relatively large-scale detection). Since it is utilizing real CDR dataset having temporal features for each cell, our work incorporates past cellular activity values into its model's learning mechanism to detect long-term attacks instead of instantaneous ones. An attack might be wrongly detected due to some instantaneous hyped value (for example, when a crowd arrives at a train station covered by a base station, the overall cellular activity will be spiked and the detector will falsely detect an attack if past cellular behavior is not considered).

III. PRELIMINARIES

A. System Model, Description of the Dataset, and Data Preprocessing

Our system model, as shown in Fig. 1, is based on the architecture of 4G LTE-A [29, Fig. 1] in which CDRs are generated at

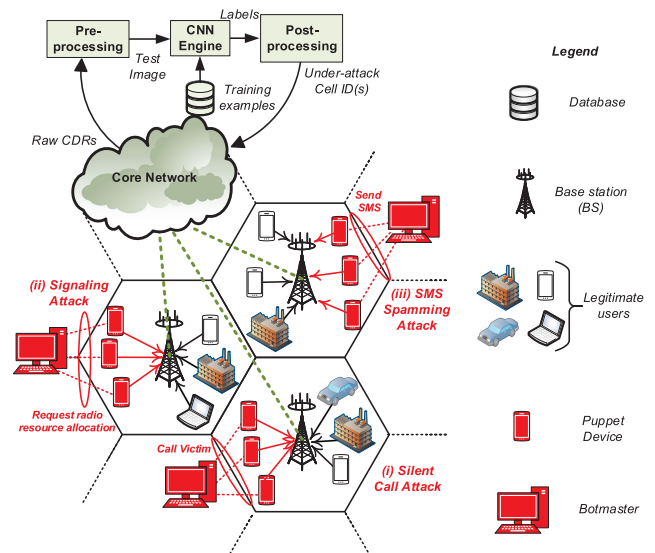


Fig. 1. System model: We assume both, legitimate devices and botnet-controlled puppet devices are trying to access the network. The compromised devices act in a certain way to cause a DDoS attack by performing: silent call, signaling, and SMS spamming attacks. The framework, installed in the CN, preprocesses the generated CDRs and creates an image processed by a deep CNN model to identify under-attack cell(s).

the CN. We assume a certain number (explained in Appendix) of botmaster-controlled puppet mobile devices in each cell along with other legitimate devices (mobile phones, devices serving a CPS, etc.). The compromised devices cause a collective DDoS attack, depriving legitimate devices from utilizing the resources, by individually performing the silent call, signaling, or SMS spamming attack.

We perform experimentation utilizing an open dataset released by Telecom Italia [30] in 2015. Temporally, it consists of over 319 million real CDRs (see Fig. 2(i) for a sample) collected over a 62-day duration (from first November 2013 to first January 2014). Spatially, the CDRs belong to 10 000 square grids or cells (used interchangeably in this article) assembled into a 100×100 grid (known as Milano grid). The grid is spread across 23.5 km^2 area of the main industrial city of Italy: Milan [30]. Each CDR log contains a timestamp value that represents beginning of a 10-min interval during which the log was registered; three activity values: outgoing calls, Internet usage, and outgoing SMS, associated with a cell; the cell's identity (ID); and some irrelevant information that we discard [30].

As the CNNs essentially accept grid-like topological data [31, Ch. 9], we select a 9×9 subgrid consisting 81 cells for our experiments, as shown in Fig. 2(ii). The small subgrid size is chosen so that the results in later sections can be easily displayed and discussed, otherwise a larger size could also be opted. The cell IDs 4259, 4456, and 5060 (marked by red in the figure) cover some of the famous locations of Milan: Bocconi university, nightlife places, and city center, respectively. Their behavioral pattern can be observed in terms of phone usage plots in [30, Fig. 7].

Our framework preprocesses raw CDRs pertaining to the selected cells for each timestamp value to create a $9 \times 9 \times 3$

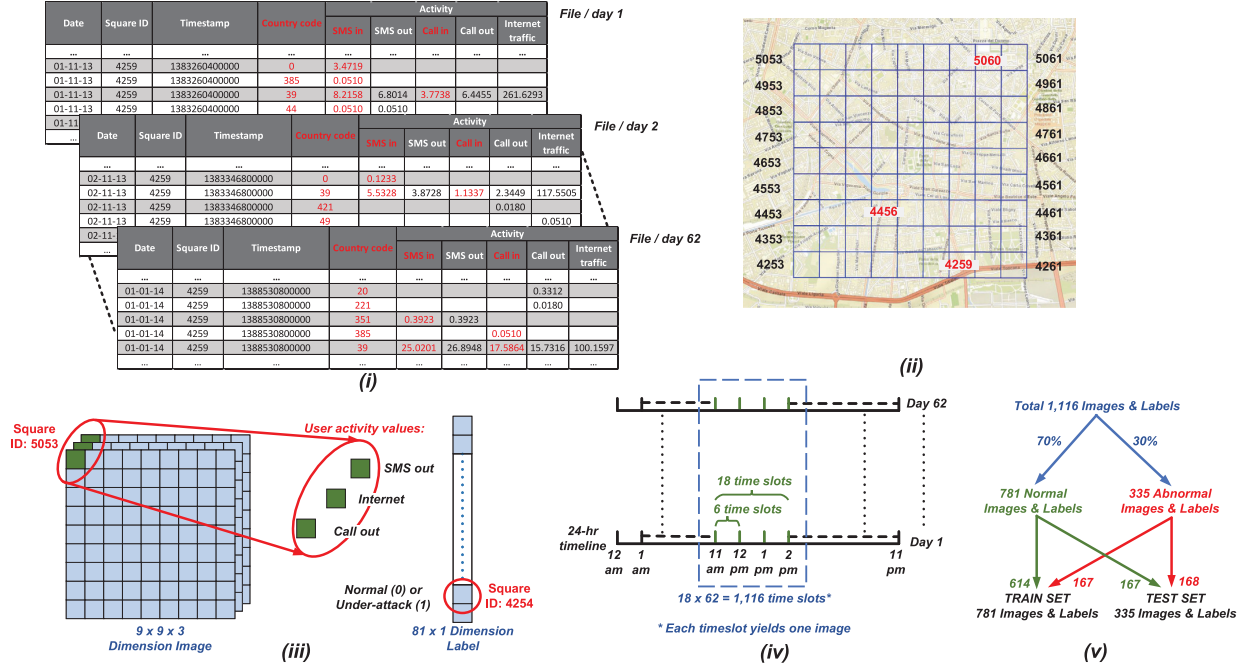


Fig. 2. Preliminaries: (i) Samples of raw CDRs, distributed among 62 files (shown as different tables), each having a single day's data. The red-highlighted values are, irrelevant to our research and, hence, discarded. (ii) Overlay of the chosen 9×9 subgrid with Milan's map, using the GPS coordinates. It represents 81 cell IDs whose CDRs are extracted to form a labeled image, as shown in (iii). Each pixel value of the image corresponds to the three user activity values of the corresponding cell ID. (iv) Illustration of the total number of time slots used for the purpose of data aggregation to create a total of 1116 images. (v) Images are combined in a way to create train and test sets.

dimension image [shown in Fig. 2(iii)] $i^{(j)} \in \mathbb{R}^{n_H^{[0]} \times n_W^{[0]} \times n_C^{[0]}}$, where the height $n_H^{[0]}$ and width $n_W^{[0]}$ represent the chosen cell IDs, the depth, or number of channels $n_C^{[0]}$ represent the three activity values, and j is the index. The image is then sent to an engine, shown in Fig. 1, running a CNN model and containing a database of past examples (images) belonging to every 10-min duration slot in a 24-h timeline. If the latest image (test example) is based on, for example, 1:10–1:20 PM slot, then the model trains on past examples belonging to the same slot and outputs the normal and underattacked cell ID(s) in the test image. Finally, the identified cell ID(s) is then passed on to the CN for the necessary actions.

B. Data Synthesis and Splitting

Data-hungry DL models require large number (hundreds or even thousands) of examples for training and we only have 62 images for every 10-min slot in a 24-h timeline, each belonging to a single day. To overcome this limitation, we adopt the method utilized in [32] by combining all the images created during a 3-h range from 11 AM–2 PM of 2 months period—6 images per hour \times 3 hours \times 62 days = 1116 images, as illustrated in Fig. 2(iv)—and considering them as images related to a single 10-min slot. These images, having 81 cells' data in each, demonstrate a normal behavior; hence, each cell is labeled as 0 (normal) in the labeled output of each image $o^{(j)} \in \mathbb{R}^{81 \times 1}$ [see Fig. 2(iii) for the illustration of corresponding label of an image].

For an image exhibiting behavior influenced by an attack, we would ideally attack an operational 4G network and notice the changes in the recorded CDRs—potentially resulting in, an

economically and legally unfeasible, network breakdown. We, hence, reserve a set of randomly chosen 335 images (30% of the total) and for each attack scenario, we utilize the set to modify the relevant user activity (call out, Internet, or SMS out) values according to Appendix to mimic the effect of the attack (silent call, signaling, or SMS spamming) on CDRs. This step is inspired from [22] in which the authors utilize simulation software to mimic the effect of different attacks and generate a synthetic CDR dataset for their experiments. Practical networks deal with normal scenarios more often as compared with the abnormal ones (anomalies), this is reflected in our model as the normal instances are chosen to be in larger quantity (70%) than the abnormal ones (30%). For the purpose of modification, we randomly choose about 50% cell IDs in each image and also change their labels to 1 (underattack).

As shown in Fig. 2(v), our train set contains 781 images (70% of the total) $I_{\text{train}} \in \mathbb{R}^{781 \times n_H^{[0]} \times n_W^{[0]} \times n_C^{[0]}}$, and their corresponding labels $O_{\text{train}} \in \mathbb{R}^{781 \times 81}$, and the test set contains the remaining ones: $I_{\text{test}} \in \mathbb{R}^{335 \times n_H^{[0]} \times n_W^{[0]} \times n_C^{[0]}}$ and $O_{\text{test}} \in \mathbb{R}^{335 \times 81}$. Out of the 781 labeled images in train set, 614 are normal and the remaining 167 are the modified images. Similarly, out of the 335 labeled images in the test set, 167 are normal and the remaining 168 are the modified images. Note, for each attack scenario, we have a separate train and test sets because of the modifications discussed previously.

C. Performance Metrics and Software Utilized

We utilize the following common metrics, which are widely used in the literature such as [33], for the performance evaluation: accuracy, error rate, precision, recall, false positive rate,

and F_1 (weighted harmonic mean of the precision and recall). We use MATLAB and Keras (Python's DL library) for the preprocessing, GPS mapping, and building the CNN models. We perform experimentation using a commercial PC (i7-7700 T CPU, Windows 10 64-b operating system, and 16 GB RAM) with an in-built GPU (NVIDIA GeForce 930MX).

IV. REALIZATION OF CNN MODELS

A. Generic Architecture

CNN has the following three fundamental building blocks.

1) **Convolution Layer:** This layer processes images or previous layer's activations $A^{[l-1]} \in \mathbb{R}^{m \times n_H^{[l-1]} \times n_W^{[l-1]} \times n_C^{[l-1]}}$, having m as the total number of images in the (train or test) dataset and l as an index of the present layer; and kernels $K^{[l]} \in \mathbb{R}^{k^{[l]} \times k^{[l]} \times n_C^{[l-1]} \times n_C^{[l-1]}}$, having $k^{[l]} \times k^{[l]} \times n_C^{[l-1]}$ as the single kernel's dimension with $k^{[l]}$ as the kernel size, and $n_C^{[l]}$ as the total number of kernels. To demonstrate a convolution layer's functionality, we focus on an example highlighted in the red box of Fig. 5. Here, the index of present (output) layer l is 2, while the previous (input) layer's index will be $l - 1 = 1$. We also consider a single image as an example; hence, $m = 1$. The input activations will then be represented as $A^{[1]} \in \mathbb{R}^{1 \times n_H^{[1]} \times n_W^{[1]} \times n_C^{[1]}}$, having $n_H^{[1]} = n_W^{[1]} = 13$ and $n_C^{[1]} = 3$. The dimension of input activations is $13 \times 13 \times 3$, as shown in the figure. Additionally, the kernels are represented as $K^{[2]} \in \mathbb{R}^{k^{[2]} \times k^{[2]} \times n_C^{[1]} \times n_C^{[1]}}$, having a kernel size $k^{[2]} = 2$ and total number of kernels $n_C^{[2]} = 8$. A single kernel's dimension is $2 \times 2 \times 3$.

The convolution layer applies convolution operation between the input activations and each kernel separately, as shown in the figure. A general convolution operation between input and a single kernel is demonstrated in [31, Fig. 9.1]. The output from each operation is then added with bias (a real number) and a nonlinear activation function called Swish is also utilized.

Swish is a gated version of sigmoid function that has some desirable properties that even the widely used and most successful activation function such as rectified linear unit (ReLU) lacks: nonmonotonicity and smoothness [34]. The inventors of Swish function claim that it yields matching or outperforming results as compared with ReLU for deeper neural networks. Mathematically, it is defined as

$$g(z) = z \times \sigma(z) \quad (1)$$

where $\sigma(z) = (1 + e^{-z})^{-1}$ is the sigmoid function.

Finally, the layer piles up each result on top of one another to create an output $A^{[l]} \in \mathbb{R}^{m \times n_H^{[l]} \times n_W^{[l]} \times n_C^{[l]}}$, which is represented as $A^{[2]} \in \mathbb{R}^{1 \times n_H^{[2]} \times n_W^{[2]} \times n_C^{[2]}}$. The height $n_H^{[2]}$ or width $n_W^{[2]}$ are computed by using

$$n_{H/W}^{[l]} = \left\lfloor \frac{n_{H/W}^{[l-1]} + 2p^{[l]} - k^{[l]}}{s^{[l]}} + 1 \right\rfloor \quad (2)$$

having $p^{[l]}$ as number of zero-padding (a technique used to insert zeros around the input image's edge to prevent shrinking of output dimension during the convolution operation [31, Sec. 9.5])

and $s^{[l]}$ as stride (distance between consecutive application of kernel on the input). For this example, the zero-padding is already previously performed [see Fig. 5 (bottom)], hence, $p^{[2]} = 0$ and $s^{[2]}$ is given as 1. By utilizing (2), we can calculate $n_H^{[2]} = n_W^{[2]} = 12$. Hence, the output's dimension will be $12 \times 12 \times 8$, which can also be observed in the figure.

In addition, batch normalization (BN) [35] technique is utilized to boost training speed and make the model robust. It is applied between convolution operation and the activation function.

2) **Pooling Layer:** This layer utilizes a max or avg function to pool maximum or average numbers, respectively, from groups of its input (and from each channel, independently) depending on the kernel size k , to generate the output volume. This reduces requirement for storing parameters and improves model's computational efficiency [31, Sec. 9.3]. If the input has $n_H \times n_W \times n_C$ dimension, the output's dimension can be derived using (2) with $p = 0$: $\lfloor \frac{n_H - k}{s} + 1 \rfloor \times \lfloor \frac{n_W - k}{s} + 1 \rfloor \times n_C$.

3) **Fully Connected Layer:** This layer has the same purpose as of a feed-forward neural network's hidden layer [32], having each neuron connected with all other previous layer's neurons.

B. Residual Network Model

The fundamental building blocks can be utilized in multiple settings (with different number of layers and the way they are linked together) to create various CNN models like residual network comprising 50 layers (ResNet-50) [36], illustrated in Fig. 3. It is one of the most advanced CNN models that we utilize in this article. Residual networks are effective in dealing with the problems encountered by a typical (very) deep neural network—gradient exploding or vanishing [31] and degradation [36] problems—by adopting residual learning in which residual blocks are extensively used.

We first elaborate functioning of a residual block using Fig. 4(top). In the figure, the information flows from input $a^{[l]}$ to the output activation $a^{[l+2]}$ via two different paths. In the downward path, known as main path, there are two parts. The information first goes through the initial part having three modules consisting of a convolution layer, BN, and a nonlinear activation function, respectively; governed by the following standard equations:

$$z^{[l+1]} = W^{[l+1]}a^{[l]} + b^{[l+1]} \quad (3)$$

$$a^{[l+1]} = g(z^{[l+1]}) \quad (4)$$

where $W^{[l+1]}$ is the weight matrix, $b^{[l]}$ is the bias vector, $g(\cdot)$ is the nonlinear activation function, $a^{[l]}$ is the input, and $a^{[l+1]}$ is the output of the first part. The BN module is added to accelerate the training.

Similarly, the modules in the second part are governed by the following equations (ignoring the other path and an addition operation):

$$z^{[l+2]} = W^{[l+2]}a^{[l+1]} + b^{[l+2]} \quad (5)$$

$$a^{[l+2]} = g(z^{[l+2]}). \quad (6)$$

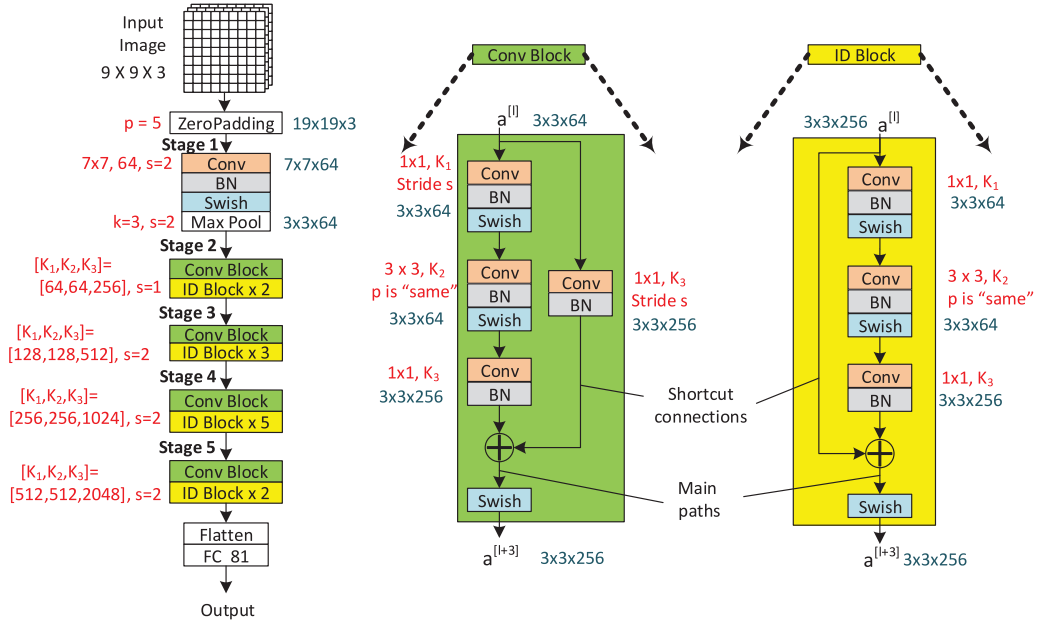


Fig. 3. Residual network architecture having 50 layers (ResNet-50). Red annotations denote the utilized hyperparameters while the blue annotations illustrate the output dimensions of the layers. Note, the displayed output dimensions in the Conv and ID blocks are for stage 2 only.

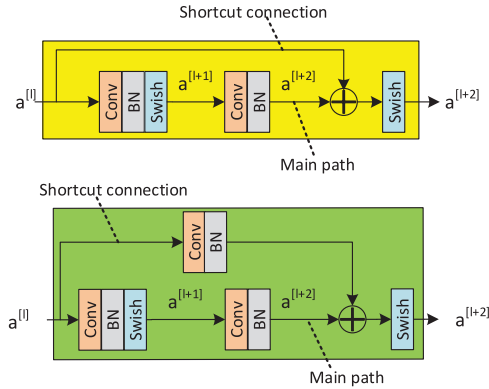


Fig. 4. Residual blocks. (Top) ID block. (Bottom) Convolutional (Conv) block.

In residual networks, $a^{[l]}$ is fast-forwarded to a deeper hidden layer in the neural network where it is added with the output of that layer before applying a nonlinear activation function. This is known as a short-cut connection, as shown in Fig. 4. Hence, (6) will be modified as follows:

$$a^{[l+2]} = g(z^{[l+2]} + a^{[l]}). \quad (7)$$

The addition of $a^{[l]}$ makes it a residual block. Here, we are assuming that the dimensions of both, input $a^{[l]}$ and $z^{[l+2]}$ (and therefore, output $a^{[l+2]}$) are same in order to perform the addition. This kind of residual block is known as ID block. If the dimensions of input ($a^{[l]}$) and output activations ($a^{[l+2]}$) do not match then a convolution layer in the shortcut connection is inserted to resize the input $a^{[l]}$ to a different dimension, so that the dimensions match up in the final addition. This type of residual block is known as Convolutional (Conv) block, as

shown in Fig. 4 (bottom). Note, we utilize residual blocks that skips three hidden layers in our paper instead of skipping two hidden layers as delineated in the figure.

In the ResNet-50 model, residual blocks are piled up on top of one another [see Stage 2–5 in Fig. 3 (left)] to grant activations of one layer to skip some layers and be directly fed to the deeper layers. During back-propagation, shortcut connections also allow a gradient to be directly back-propagated to the previous layers. As can be seen in the figure that the input image having dimension $9 \times 9 \times 3$ is zero-padded with padding $p = 5$ to have an output volume with dimension $19 \times 19 \times 3$ [(2) can be utilized in calculating the output dimension of various layers]. The resultant volume is then passed to Stage 1 having a convolution layer with kernel size $k = 7$, total number of kernels $n_C = 64$, and stride $s = 2$; that converts the dimension to $7 \times 7 \times 64$. Finally, pooling layer (Max Pool) having $k = 3$ and $s = 2$ yields the output volume with dimension $3 \times 3 \times 64$.

For Stage 2, middle part of Fig. 3 having Conv block will have an input dimension of $3 \times 3 \times 64$ from the previous layer. The main path contains three parts. The first part has convolution layer having $k = 1$, $n_C = K_1 = 64$, and $s = 1$. It outputs volume with same dimensions as that of the input. The convolution layer in the second part also results output with same dimension as that of the input, because it is utilizing “same” convolution (in which padding is set so that the output’s dimension remains same as that of the input). The third part having a convolution layer with $k = 1$, $n_C = K_3 = 256$, and $s = 1$ will transform the input’s dimension from $3 \times 3 \times 64$ to $3 \times 3 \times 256$. Finally, convolution layer in the shortcut connection, which has input volume of dimension $3 \times 3 \times 64$, scales up the input’s dimension to $3 \times 3 \times 256$ by utilizing the following parameter values $k = 1$, $n_C = K_3 = 256$, and $s = 1$. The outputs from both

convolution layers (one in the shortcut connection and the other in third part of the main path) can be added as they are now compatible: have same dimensions.

The ID blocks of Stage 2 have similar function as of the abovementioned Conv block, with the exception of the shortcut connection's design that does not have any layer in it. This is because the input of the ID blocks has same dimension as of the output of convolution layer in its third part: $3 \times 3 \times 256$; hence, convolution layer is not needed in the shortcut connection.

The rest of the stages (Stage 3–5) follow a similar pattern as mentioned above and ultimately yield a resultant volume of dimension $1 \times 1 \times 32$. It is then flattened in the form of an array and passed on to a final fully connected layer (50th layer) to be processed as a 81×1 dimension output vector carrying normal and underattack cell IDs. The hyperparameters used in our model and the abovementioned dimensions of various layers from input layer to the layers utilized in Stage 2 can be found in Fig. 3 in the form of red and blue annotations, respectively. A softmax function [31, Sec. 4.1] is typically utilized in the output layer for a multiclass classification problem; however, since we are dealing with multilabel classification problem, we use binary cross entropy loss function.

C. DRC Model

Keeping in view the relatively lesser input image dimensions ($9 \times 9 \times 3$, i.e., 81 pixels) that we are dealing with, we design a relatively simple, six-layer model named as the DRC model. It is built from the fundamental (convolution, pooling, and fully connected) layers extensively described in Section IV-A; and is inspired from the designs of classical models like VGG [37], AlexNet [38], and LeNet-5 [39]. It is illustrated in Fig. 5 (Bottom).

The model takes an input image having dimension $9 \times 9 \times 3$ and expands it by padding zeros with $p = 2$ to yield a volume with dimension $13 \times 13 \times 3$. Then, the model passes the volume through a convolution layer to transform its dimension to $12 \times 12 \times 8$ (see Section IV-A1 for complete details of this step). Next, a max-pooling layer is utilized with $k = s = 2$. We can apply (2) to get the output volume's dimension as $6 \times 6 \times 8$. In a similar manner, the DRC model then passes the resultant volume through a series of convolution and pooling layers (Conv2, MaxPool2, Conv3, and MaxPool3), delineated in the figure. The resultant volume is finally flattened and passed through the two fully connected layers (FC1 and FC2) to give a 81×1 dimension output vector (by utilizing binary cross entropy loss function), having identification of normal and underattack cells.

V. EXPERIMENTAL RESULTS AND PERFORMANCE EVALUATION

We utilize test set $\{I_{\text{test}}, O_{\text{test}}\}$ (containing 335 images and their corresponding labels) for the performance evaluation of our models under various attack scenarios, and report the results in Figs. 6 and 7. Overall, for all the attack scenarios except the blended attack, our DRC model surpassed ResNet-50 model in terms of all the performance metrics, as evident in Fig. 6. Additionally, the maximum difference in the performance between

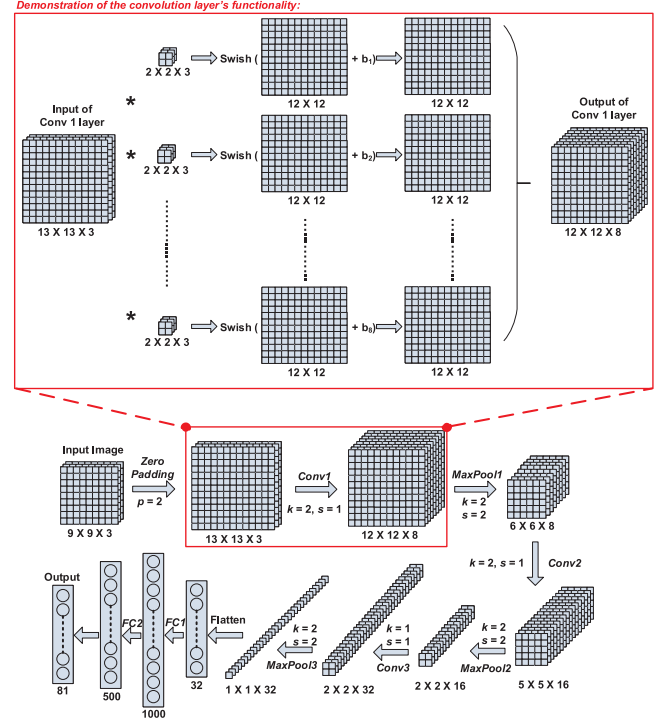


Fig. 5. (Bottom) DRC model. (Top) The red box demonstrates the operations of a convolution layer utilized in our DRC model.

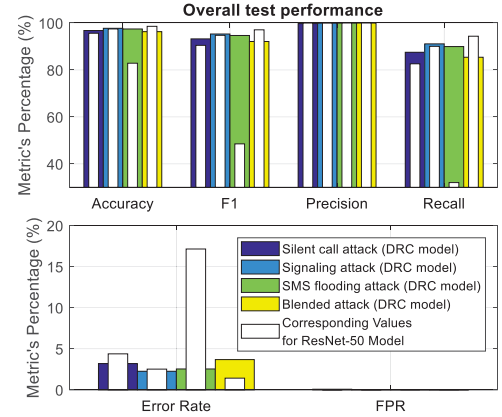


Fig. 6. Overall test performance of DRC and ResNet-50 models, utilizing 100 epochs.

the two models is observed during the SMS flooding attack for which ResNet-50 model performed poorly; while for signaling attack both models performed in a similar fashion.

Due to the limited space, we only illustrate accuracies yielded from the various attack scenarios as heatmaps in Fig. 7. Each 9×9 heatmap from (a)–(e) of the figure contains accuracy values of the corresponding cells in Fig. 2(ii). We can observe that the results are varied across the spectrum of cells depending on the individual cellular activity levels and the consequent learning of the model. Lowest accuracies for each attack scenario are marked by black ovals in the figure. We can observe that our DRC model yielded more than 91% accuracy for every cell in the subgrid under every attack scenario (green subgrids).

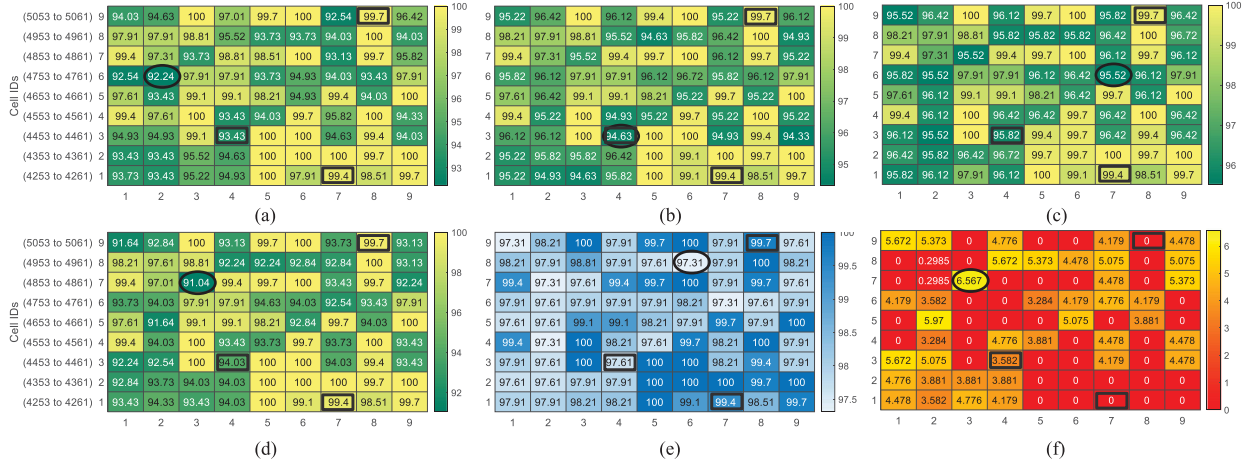


Fig. 7. Accuracy distributions of our models under various attack scenarios. (a)–(d) Results of our DRC model. (e) Performance of ResNet-50 model for blended attack scenario, in which it outperformed our model. (f) Improvements we achieved with ResNet-50 model under the blended attack scenario. (a) Silent call attack scenario: Deep Rudimentary CNN (DRC) model's accuracy distribution. (b) SMS flooding attack scenario: DRC model's accuracy distribution. (c) Signaling attack scenario: DRC model's accuracy distribution. (d) Blended attack scenario: DRC model's accuracy distribution. (e) Blended attack scenario: ResNet-50 model's accuracy distribution. (f) Blended attack scenario: Difference between ResNet-50 and DRC models.

Since, the ResNet-50 model has a superior performance under blended attack scenario, we also demonstrate accuracy heatmap (blue subgrid) for our ResNet-50 model and the improvement (red subgrid), it achieved as compared with the DRC model in Fig. 7(e) and (f), respectively. The higher performance can also be judged from the clear difference in minimum accuracy values yielded by both models (annotated with black ovals) in Fig. 7(d) and (e): 91% for the DRC model and 97% for the ResNet-50 model. Additionally, it can also be observed in Fig. 7(d)–(f) that the worst performing cell 4855 (row 7, column 3) with 91% accuracy under the blended attack scenario improves to have 97.6% by applying the ResNet-50 model.

Interestingly, it can also be noted that the cells covering the Bocconi university (cell ID 4259: row 1, column 7) and the city center (cell ID 5060: row 9, column 8) have steady and high accuracy values (highlighted in black rectangles in the figure) throughout all the attack scenarios and for both models. This might be because of the relative high user activities in these popular areas during the selected (lunch) timings (from 11 AM to 2 PM), for which both models were able to easily distinguish the hidden pattern and, hence, detected normal and underattacked cell(s) with high accuracies; in contrast to the relatively low accuracy values for the cell covering nightlife places (ID 4456: row 3, column 4) that has relatively low user activity values during the selected timings.

VI. CONCLUSION

Our framework achieved higher than 91% normal and underattack cell detection accuracy by utilizing the DRC model for silent call, signaling, and SMS flooding attacks that target a cellular network to cause DDoS to the cellular connectivity-dependent legitimate devices, including the ones utilized in the CPSs, as described in Section I. The framework also attained higher than 97% accuracy for a more sophisticated blended attack, in which each puppet device performed all the three

attacks, by using ResNet-50 model. Our results suggested that for an individual attack, where its effect was limited to a single user activity value modification in the CDRs, our framework employing DRC model can more effectively detect the cell ID(s) under attack as compared with utilizing a ResNet-50 model. While for the blended attack ResNet-50 model can yield better accuracy due to its very deep neural network design that can effectively learn the intricate structure in the dataset.

Upon detection, the information can then be sent from our coarse-grained analysis framework to the CPSs to trigger defensive/mitigative measures and can also be utilized to further perform fine-grained analysis [10, Sec. VI.C.]. For example, by acquiring more denser and richer underattack cell's data including every user equipment's data, and feeding them to a feed forward deep neural network. This would heavily aid in identifying the bots/adversary devices within a short time, such as in minutes—it usually takes a month for most organizations to identify and clear the puppet devices [11, Fig. 18]. Our work can naturally fit to support mobile edge computing (MEC) paradigm [40] in cellular networks having MEC servers geographically located across the network and each server, colocated with a base station, monitoring cellular activity of a subgrid, and running our proposed framework. The benefit of such setting resides in dividing computation-intensive tasks across the network (among MEC servers), easing computation and storage for the CN. By leveraging voice CDRs, our work can be extended to detect overcharging attacks that can potentially be engineered to launch DDoS attacks [25].

Our robust framework can perform simultaneous analysis on multiple cells, depending on the size of subgrid, due to the inherent utilization of CNN architecture. It can be scaled-up to consider a larger subgrid; however, the computation requirements need to be investigated keeping in view the online and offline settings. As we had a limited dataset, we combined 3 h data of 62 days and considered it as past data belonging to

a 10-min slot (explained thoroughly in Section III); in practice, historical CDR dataset was maintained for record-keeping within the cellular network and may easily be acquired. They might also yield improved results as the model would learn from the data containing same temporal characteristics (one 10-min slot instead of 18 slots).

Since many devices, including the ones utilized in CPSs, depend on cellular infrastructure and its services for connectivity—for example, IoT devices use voice services [25], WSN devices rely on Internet services [28], and M2M communication network devices utilize SMS services [15]—our research is compatible as our framework leverages each service's usage data, and has solid applications in their security and earlier detection of DDoS attacks against them.

In conclusion, this is a pioneering study that investigated the application of CNNs for the cellular network's security in a coarse-grained manner to detect various attacks that lead to a DDoS (voice, Internet, and SMS) and achieved more than 91% accuracy—contributing to resolve an open issue of DDoS attack mitigation in cellular networks [9]. Besides the primary subscriber devices, our study has solid implications in securing cellular-dependent CPS devices (utilized in vertical industries and critical infrastructures) against cellular DDoS attacks that could serve as a beachhead or smoke screen to attack the CPS infrastructure and disrupt its services.

APPENDIX EMULATING EACH ATTACK'S EFFECT

Since the dataset provider has applied checks to keep user privacy and hid details about the number of devices in each square grid, we estimate the number using an indirect manner: $7157 \text{ residents/km}^2$ (Milan's population density in 2014 [41]) $\times 0.235 \text{ km}^2$ (area of a square grid [30]) $\times 1.509$ (Italy's mobile cellular subscriptions per capita in 2014 [42]) $\times 34\%$ (market share of Telecom Italia [30]) ≈ 863 mobile devices per cell of Milano grid. 6% botnet-infected devices, according to [16], are enough to cause DDoS attack in 4G networks: we assume about 52 puppet devices. In the following, we calculate the times we have to increase or decrease the relevant user activity value to emulate the effect of corresponding attack scenario:

1) *Silent Call Attack*: A typical user talks 761.5 min/30 days according to [16], resulting in 0.302 min/10-min duration (assuming 8 AM–10 PM: 14 hours per day, in which most calls happen). The ratio of simultaneously active over average subscribers in a typical cell is 1:33 [16], which leads to ≈ 26 active subscribers in the square grid. In normal situation, the total number of outgoing calls in a 10-min slot are: $10/0.302 \times 26 \approx 861$ calls or CDRs generated. If we assume 26 botnet-controlled devices performing silent call attack, they will generate only 26 CDRs and meanwhile the legitimate devices are denied of the service. Consequently, about $861/26 \approx 33.1 \times$ lesser CDRs/user activity will be registered.

2) *Signaling Attack*: A malicious device can initiate up to eight dedicated bearers and for each, three bearer activation, and three bearer deactivation (signaling) messages are generated in 2 min [26]; aggregating to 240 messages per device in 10-min.

Since in our dataset, the Internet activity CDR is recorded each time a device starts or ends an Internet connection [30], we assume each of the 240 messages generates a CDR. The botnet-controlled 52 devices can then generate 12 480 CDRs. We can approximately calculate number of CDRs generated by a typical (normal) device by analyzing HTTP requests versus number of mobile devices plot in [43, Fig. 1(left)]. From the linear trend in the plot, we can process the middle point as $10^5 \text{ (requests)} / 10^3 \text{ (devices)} = 100 \text{ requests/device over 7 days} \approx 0.3404 \text{ CDRs per device in 10 min}$ (considering 14 hours per day, as mentioned previously, and also each HTTP request can generate two CDRs, each upon device connection and disconnection). Hence, a total of 12 756 CDRs (12 480 CDRs by 52 malicious devices + 276 CDRs by remaining 811 normal devices) will be generated under the attack scenario; while only 294 CDRs (0.3404×863 users) will be generated in a cell under normal condition. Overall, about $12\,756/294 \approx 43.3 \times$ more Internet CDRs/user activity will be logged.

3) *SMS Spamming Attack*: Android application allows max. 30 SMS/30-min to be sent while *HackFacebook* application allows at least 1002 SMS/30-min [27]. Let's say a typical user sends ten SMS/day; a total of 103 SMS/10-min (assuming 14-hours a day, as mentioned previously) can be sent by all devices under normal conditions and a total of 17 465 SMS/10-min can be sent under the attack scenario (17 368 SMS by 52 malicious devices + 97 SMS by the remaining 811 devices). In total, about $17\,465/103 \approx 169.5 \times$ more SMS CDRs or user activity will be registered.

4) *Blended Attack*: In this extreme case, we assume each malicious device is performing all three attacks. Hence, we modify the values of all user activities accordingly.

REFERENCES

- [1] A. Humayed, J. Lin, F. Li, and B. Luo, "Cyber-physical systems security—A survey," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 1802–1831, Dec. 2017.
- [2] H. Song, D. Rawat, S. Jeschke, and C. Brecher, *Cyber-Physical Systems: Foundations, Principles and Applications*, Boston, MA, USA: Academic Press, 2016.
- [3] 5G-PPP. White paper on 5G and the Factories of the Future, Oct. 2015. [Online]. Available: <https://5g-ppp.eu/wp-content/uploads/2014/02/5G-PPP-White-Paper-on-Factories-of-the-Future-Vertical-Sector.pdf>
- [4] M. Wollschlaeger, T. Sauter, and J. Jasperneite, "The future of industrial communication: Automation networks in the Era of the Internet of Things and industry 4.0," *IEEE Ind. Electron. Mag.*, vol. 11, no. 1, pp. 17–27, Mar. 2017.
- [5] C. Tranoris, S. Denazis, L. Guardalben, J. Pereira, and S. Sargento, "Enabling cyber-physical systems for 5G networking: A case study on the automotive vertical domain," in *Proc. IEEE/ACM 4th Int. Workshop Softw. Eng. Smart Cyber-Phys. Syst.*, 2018, pp. 37–40.
- [6] [Online]. Available: <https://5g-ppp.eu/white-papers/>
- [7] S. Jeschke, C. Brecher, H. Song, and D. Rawat, *Industrial Internet of Things: Cybermanufacturing Systems*. Cham, Switzerland: Springer, 2017.
- [8] M. Cosovic, A. Tsitsimelis, D. Vukobratovic, J. Matamoros, and C. Anton-Haro, "5G mobile cellular networks: Enabling distributed state estimation for smart grids," *IEEE Commun. Mag.*, vol. 55, no. 10, pp. 62–69, Oct. 2017.
- [9] S. Mavoungou, G. Kaddoum, M. Taha, and G. Matar, "Survey on threats and attacks on mobile networks," *IEEE Access*, vol. 4, pp. 4543–4572, 2016.
- [10] L. He, Z. Yan, and M. Atiquzzaman, "LTE/LTE-A network security data collection and analysis for security measurement: A survey," *IEEE Access*, vol. 6, pp. 4220–4242, 2018.

- [11] Verizon. 2018 data breach investigations report, 2018. [Online]. Available: https://enterprise.verizon.com/resources/reports/DBIR_2018_Report.pdf
- [12] I. Kolochenko, DDos attacks: A perfect smoke screen for APTs and silent data breaches, Sep. 2015. [Online]. Available: <https://www.csoonline.com/article/2986967/ddos-attacks-a-perfect-smoke-screen-for-apt-and-silent-data-breaches.html>
- [13] G. Tu, C. Li, C. Peng, and S. Lu, "How voice call technology poses security threats in 4G LTE networks," in *Proc. IEEE Conf. Commun. Netw. Secur.*, 2015, pp. 442–450.
- [14] A. Gupta, T. Verma, S. Bali, and S. Kaul, "Detecting MS initiated signaling DDos attacks in 3G/4G wireless networks," in *Proc. Int. Conf. Commun. Syst. Netw.*, 2013, pp. 1–6.
- [15] I. Murynets and R. P. Jover, "Anomaly detection in cellular machine-to-machine communications," in *Proc. IEEE Int. Conf. Commun.*, 2013, pp. 2138–2143.
- [16] M. Khosroshahy, D. Qiu, and M. K. M. Ali, "Botnets in 4G cellular networks: Platforms to launch DDos attacks against the air interface," in *Proc. Int. Conf. Sel. Topics Mobile Wireless Netw.*, 2013, pp. 30–35.
- [17] H. Zhang, Z. Wang, and Q. Du, "Social-aware D2D relay networks for stability enhancement: An optimal stopping approach," *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 8860–8874, Sep. 2018.
- [18] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.
- [19] C. S. Wickramasinghe, D. L. Marino, K. Amarasinghe, and M. Manic, "Generalization of deep learning for cyber-physical system security: A survey," in *Proc. 44th Annu. Conf. IEEE Ind. Electron. Soc.*, 2018, pp. 745–751.
- [20] Department of Defense, USA. *Summary of the 2018 DoD Artificial Intelligence Strategy*, Feb. 2019. [Online]. Available: <https://media.defense.gov/2019/Feb/12/2002088963/-1/-1/1/SUMMARY-OF-DOD-AI-STRATEGY.PDF>
- [21] E. Ernst, R. Merola, and D. Samaan, "The economics of artificial intelligence: Implications for the future of work," *International Labour Organization, Geneva, Switzerland, Res. Paper Series*, Oct. 2018.
- [22] S. Papadopoulos, A. Drosou, and D. Tzovaras, "A novel graph-based descriptor for the detection of billing-related anomalies in cellular mobile networks," *IEEE Trans. Mobile Comput.*, vol. 15, no. 11, pp. 2655–2668, Nov. 2016.
- [23] N. Ruan *et al.*, "A traffic based lightweight attack detection scheme for VoLTE," in *Proc. IEEE Global Commun. Conf.*, 2016, pp. 1–6.
- [24] S. Papadopoulos, A. Drosou, I. Kalamaras, and D. Tzovaras, "Behavioural network traffic analytics for securing 5G networks," in *Proc. IEEE Int. Conf. Commun. Workshops*, 2018, pp. 1–6.
- [25] T. Xie, C. Li, J. Tang, and G. Tu, "How voice service threatens cellular-connected IoT devices in the operational 4G LTE networks," in *Proc. IEEE Int. Conf. Commun.*, 2018, pp. 1–6.
- [26] R. Bassil, A. Chehab, I. Elhajj, and A. Kayssi, "Signaling oriented denial of service on LTE networks," in *Proc. ACM Int. Symp. Mobility Manage. Wireless Access*, 2012, pp. 153–158.
- [27] G.-H. Tu, C.-Y. Li, C. Peng, Y. Li, and S. Lu, "New security threats caused by IMS-based SMS service in 4G LTE networks," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2016, pp. 1118–1130.
- [28] J.-H. Bang, Y.-J. Cho, and K. Kang, "Anomaly detection of network-initiated LTE signaling traffic in wireless sensor and actuator networks based on a Hidden semi-Markov model," *Comput. Secur.*, vol. 65, pp. 108–120, Mar. 2017.
- [29] M. S. Parwez, D. Rawat, and M. Garuba, "Big data analytics for user-activity analysis and user-anomaly detection in mobile wireless network," *IEEE Trans. Ind. Informat.*, vol. 13, no. 4, pp. 2058–2065, Aug. 2017.
- [30] G. Barlacchi *et al.*, "A multi-source dataset of urban life in the city of Milan and the province of trentino," *Sci. Data*, vol. 2, no. 150055, pp. 1–15, 2015.
- [31] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, Cambridge, MA, USA: MIT Press, 2016.
- [32] B. Hussain, Q. Du, and P. Ren, "Deep learning-based big data-assisted anomaly detection in cellular networks," in *Proc. IEEE Global Commun. Conf.*, 2018, pp. 1–6.
- [33] B. Hussain, Q. Du, and P. Ren, "Semi-supervised learning based big data-driven anomaly detection in mobile wireless networks," *China Commun.*, vol. 15, no. 4, pp. 41–57, Apr. 2018.
- [34] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for Activation Functions," Oct. 2017, *arXiv:1710.05941v1*.
- [35] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.
- [36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 770–778.
- [37] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Representations*, 2015, pp. 1–14.
- [38] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Adv. Neural Inform. Process. Syst.*, 2012, pp. 1097–1105.
- [39] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
- [40] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: the communication perspective," *IEEE Commun. Surv. Tut.*, vol. 19, no. 4, pp. 2322–2358, 4th Quart., 2017.
- [41] [Online]. Available: <https://www.citypopulation.de/php/italy-lombardia.php?cityid=015146>
- [42] [Online]. Available: <https://data.worldbank.org/indicator/IT.CEL.SETS.P2?view=map&year=2014>
- [43] X. An and G. Kunzmann, "Understanding mobile Internet usage behavior," in *Proc. IFIP Netw. Conf.*, 2014, pp. 1–9.



Bilal Hussain (Student Member, IEEE) received the B.E. degree (first-class Hons.) in electrical engineering from Bahria University, Islamabad, Pakistan, in 2010, and the M.Sc. degree in information and communications engineering from the University of Leicester, Leicester, U.K., in 2011. He is currently working toward the Ph.D. degree in information and communications engineering from Xi'an Jiaotong University, Xi'an, China.

His broader research interests include applications of artificial intelligence and big data analytics in wireless communication systems (6G/5G mobile networks), mobile edge and fog computing, and cyber-physical systems security.



Qinghe Du (Member, IEEE) received the B.S. degree in information engineering and M.S. degree in information and communications engineering from Xi'an Jiaotong University, China, in 2001 and 2004, respectively, and the Ph.D. degree in computer engineering from Texas A&M University, College Station, TX, USA, in 2010.

He is currently a Professor with the School of Information and Communications Engineering, Xi'an Jiaotong University. He has authored or coauthored more than 100 technical papers. His research interests include mobile wireless communications and networking with emphasis on security assurance in wireless transmissions, AI-empowered networking technologies, 5G networks and its evolution, cognitive radio networks, industrial Internet, blockchain and its applications, Internet of Things, etc.

Dr. Du was the recipient of the Best Paper Award in IEEE GLOBECOM 2007 and IEEE COMCOMAP 2019, respectively, and the Best Paper Award of *China Communications* in 2017. He was or is an Associate Editor for the IEEE COMMUNICATIONS LETTERS and an Editor for *KSII Transactions on Internet and Information Systems*. He was/is a Technical Program Co-Chair for IEEE International Congress on Cognitive Computing Workshop on Internet of Things (IoT) 2013–2017, a Track Co-Chair for ICI 2015–2019, and the Publicity Co-Chairs for the IEEE International Conference on Communications (ICC) 2015 Workshop on IoT/CPS-Security, IEEE GLOBECOM 2011, International ICST Wireless Internet Conference 2011, and ICST QShine 2010. He is/was the Technical Program Committee Members for many world-renowned conferences including IEEE INFOCOM, GLOBECOM, ICC, International Symposium on Personal, Indoor and Mobile Radio Communications, Vehicular Technology Conference, etc.



Bo Sun (Member, IEEE) received the B.S. degree in information engineering and M.S. degree in information and communications engineering both from Xi'an Jiaotong University, Xi'an, China, in 1999 and 2002, respectively.

He is currently a Senior Specialist in Wireless Communications Technology with ZTE Corporation, Shenzhen, China. His research interests include mobile wireless communications and networking transmissions, short-range wireless communication technologies, WLAN, edge computing and AI-empowered wireless communications, 5G and future networking technologies, etc.

Dr. Sun is very active in IEEE wireless standard development. He is currently the Chair of IEEE 802.11 TGbd, also the PHY adhoc Co-Chair of IEEE 802.11 TGax.



Zhiqiang Han received the B.S. degree in industry design and M.S. degree in information and communication engineering from Sichuan University, Chengdu, China, in 2006 and 2010, respectively.

He is currently a Technical Pre-Research Expert of ZTE Corporation, Shenzhen, China. His research interest include wireless local area network, edge computing, Internet of Things, 5G networks, etc.