Probability in the Engineering and Informational Sciences, **30**, 2016, 514–531.
doi:10.1017/S0269964816000140

DETECTING NETWORK-UNFRIENDLY MOBILES WITH THE RANDOM NEURAL NETWORK

Omer H. Abdelrahman
Intelligent Systems and Networks Group
Department of Electrical and Electronic Engineering
Imperial College London, SW7 2BT
UK
E-mail: o.abd06@imperial.ac.uk

Mobile networks are universally used for personal communications, but also increasingly used in the Internet of Things and machine-to-machine applications in order to access and control critical services. However, they are particularly vulnerable to signaling storms, triggered by malfunctioning applications, malware or malicious behavior, which can cause disruption in the access to the infrastructure. Such storms differ from conventional denial of service attacks, since they overload the control plane rather than the data plane, rendering traditional detection techniques ineffective. Thus, in this paper we describe the manner in which storms happen and their causes, and propose a detection framework that utilizes traffic measurements and key performance indicators to identify in real-time misbehaving mobile devices. The detection algorithm is based on the random neural network which is a probabilistic computational model with efficient learning algorithms. Simulation results are provided to illustrate the effectiveness of the proposed scheme.

1. INTRODUCTION

In mobile communications, signaling refers to the message exchanges that occur between mobile devices and a network to setup, maintain and release connections. It provides basic functions such as mobility management, radio resource control (RRC), authentication, accounting, etc., which form the control plane of the network. Recently, the number of mobile devices and applications requiring constant access to the Internet has been growing exponentially, placing greater demands on the data and signaling infrastructures of service providers. While operators benefit from such growth in data and billing-related signaling, since it directly correlates to their increased revenues and can be handled effectively through capacity engineering, they are struggling with RRC-based *signaling storms* caused by malfunctioning applications, malware and malicious behavior. Such storms can cause disruption in the access to the infrastructure, through sudden overload in the signaling backbone of mobile networks [4,40] and potentially the wireless bandwidth of users, and may

 0269-9648/16 $25.00

also deplete the battery power of mobile devices [21] and increase the energy consumption of base stations and core networks (CNs) [36].

Mobile networks are also increasingly used in the Internet of Things (IoT) and machine-to-machine (M2M) applications in order to access and control critical industrial and commercial environments. As such they are a key part of our critical infrastructure which can be compromised by these signaling storm effects. On the other hand, IoT and M2M applications could also be responsible for degrading the performance of mobile networks, due to the large number of devices to be supported that may act in a synchronized manner inherent in signaling storms.

These and other challenges [5] require the development of new technical solutions to make mobile networks more resilient and reliable. This is particularly the case with signaling storms which are difficult to detect using traditional denial of service (DoS) defense mechanisms [3,35,47,51,52], since they overload the control plane while leaving the data plane mostly unaffected.

Thus, this paper introduces a signaling storm detection system based on the random neural network (RNN) introduced by Gelenbe in [23]. The RNN is a probabilistic computational model which was inspired by the spiking behavior of neurons, and which has a well-developed mathematical theory [23,24,31] and efficient learning algorithms for recurrent networks [25,33,38]. The RNN has been successfully applied to several problems in engineering and information sciences, including pattern recognition [9,10,30,34], classification [32], image/video processing and compression [15–17,29,37], DoS attack detection [35,51,52], and others that can be found in numerous reviews on the subject [26,27,61].

1.1. Contributions of the Paper

In this paper, we develop a supervised RNN-based approach for detecting mobile devices that generate excessive RRC signaling, without directly monitoring the control plane itself. In contrast to signaling-based techniques [19,28] which can be more effective but costly, the present approach intercepts packets at the edge of the mobile network using standard monitoring technologies. This offers the advantages of not requiring to decode lower radio related layers, lack of network encryption, and fewer number of nodes to monitor [60]. Moreover, the algorithm relies mainly on timestamps and packet header information to classify users, and does not require knowledge of the application generating a packet nor its service type, thus eliminating the need to use a commercial deep packet inspection tool which may result in considerable overhead. It also interacts with existing network management systems to reduce computational overhead, storage requirements and false alarm rate. The use of a supervised learning RNN is motivated by its capability of classifying known patterns such as signaling storms whose characteristics and root causes are well understood [4,5,21,40], and also its previous success in detecting traditional DoS attacks in the Internet [35,52].

The rest of this paper is organized as follows. We discuss the characteristics and causes of signaling storms, and review related work in Section 2. Section 3 provides a brief summary of the RNN model as applied to our problem of distinguishing between normal and misbehaving mobile devices. The core of the detection technique is presented in Section 4, including the classification process, the choice of input features, and the parameters that can influence the performance of the algorithm. In Section 5, we evaluate our detection mechanism using data generated by a detailed discrete-event mobile network simulator [39,40]; we describe the user and attack models, and present experimental results. Finally, we summarize our findings in Section 6.

https://doi.org/10.1017/S0269964816000140 Published online by Cambridge University Press

2. BACKGROUND

2.1. The RRC Protocol

In mobile networks, the RRC protocol is used to manage resources in the radio access network (RAN). It performs functions such as setup, configuration, maintenance and release of radio bearers between the user equipment (UE) (i.e., mobile device) and the network, and carries non-access stratum signaling to the CN for mobility, session and identity management. In order to perform these functions, the RRC protocol associates to each UE a state machine in which different states have different amounts of radio resources and power consumption levels. State promotions occur when a UE sends or receives traffic, while state demotions are triggered by inactivity timers. The state machine is designed to allow efficient use of available spectrum and battery power of UEs, by freeing up resources when they are not being used, but the cost in terms of signaling load is paid during state transitions.

Typically, there are at least two RRC states: *idle* and *connected*. In the idle mode, the UE does not have a signaling connection with the network, consumes negligible amount of energy, and its location in not known precisely by the network. Thus, traffic destined for a UE in idle mode will require paging in order to locate the UE at the cell level. In the connected mode, the UE has a signaling connection, its location is known at the level of a single cell, and it can communicate at a data rate which depends on traffic load, quality of service requirements, mobility, etc. There can be multiple sub-states within the connected mode, depending on the mobile technology employed and the specific implementation of each network operator.

Figure 1 shows two possible implementations of the RRC state machines in 3G/UMTS and 4G/LTE systems, and the typical number of signaling messages exchanged within the RAN for each transition. One can observe that promotions from idle to connected are quite expensive in terms of signaling, thus motivating the introduction of sub-states in the connected mode. In UMTS, there are usually three sub-states: a low-energy cell_PCH state which allows the UE to stay in the connected mode without being able to transfer data, a low bandwidth cell_FACH state, and a high bandwidth cell_DCH state. In LTE, the UE has the ability to go into short and long discontinuous reception (DRX) states while in the connected mode, where it sleeps most of the time and periodically wakes up to check if there is data to be transferred, with longer sleep periods in long DRX than in short DRX.

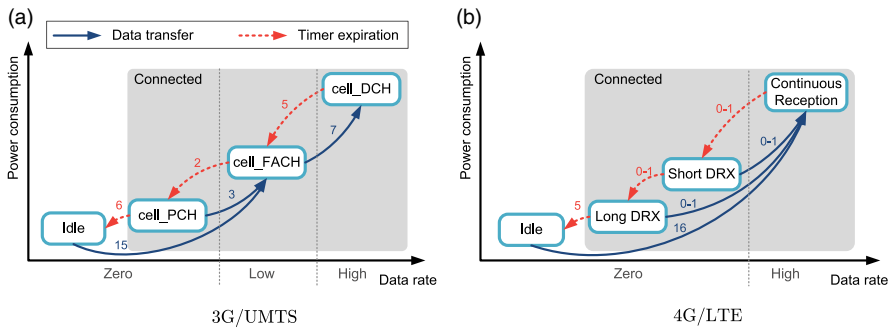

FIGURE 1. RRC state machines in UMTS and LTE, where the values on the arrows indicate the typical number of signaling messages exchanged within the RAN for each transition.

https://doi.org/10.1017/S0269964816000140 Published online by Cambridge University Press

2.2. Causes of Signaling Storms

The vulnerability of mobile networks to deliberate signaling DoS attacks is not new, and much work has been done to identify intrinsic characteristics of the networks' normal operations that could be exploited, for example, paging [58], service requests [62] and RRC [45,57]. However, such threats remained largely unsubstantiated, due to (i) the lack of financial incentives for cyber-criminals to bring down the infrastructure that they use to launch profitable attacks, and (ii) the difficulty of spreading malware onto a sizeable number of devices before the proliferation of application marketplaces. This situation changed with the advent of smart devices and applications, and many operators have since experienced unintentional signaling storms that have the same effect as a DoS attack. Such storms occur when a large number of mobile devices make successive connection requests that time-out because of inactivity, triggering repeated RRC signaling to allocate and de-allocate radio channels and other resources in the network.

Poorly designed mobile applications are one of the most common triggers of signaling overloads [7] that lead to performance degradation and even network outages [18,22]. Such applications constantly poll the network even when users are inactive in order to enable continuous connectivity [50], user behavior measurement and advertisements [14]. A common issue with those "chatty", signaling-intensive applications is that developers are not familiar with the control plane of mobile networks, which prompted the mobile industry to promote best practices for developing network-friendly applications [8,20,41,43]. Similar problems have been reported with M2M systems which periodically transmit small amounts of data [44,59], motivating the development of new standards for M2M communications [1].

However, industry guidelines for developers are not sufficient, since well-designed applications could also trigger a storm, when an unexpected event occurs in the Internet. Examples of such events include outages in cloud services [12,55] and VoIP peer-to-peer networks [13], where a large number of mobile devices attempt to recover connectivity to the application servers, generating significantly more keep-alive messages [6] and an unexpectedly high signaling load in the process.

In addition, signaling storms may occur as a by-product of malicious activity that is not intended to cause a signaling DoS incident. For example, unwanted traffic in the Internet [56] (e.g., port scans, spam campaigns, etc.) can create a storm upon reaching a mobile network, which is possible because many operators [54,63] allow mobiles to be probed from the Internet, by either assigning them public IP addresses, allowing IP spoofing, or permitting device-to-device probing within the network. Large-scale mobile malware infections may also trigger a storm, if the malware exhibit frequent communications as in premium SMS diallers, spammers and adware which are among the top encountered threats on smart devices [48]. This is confirmed by a recent analysis of mobile subscribers' traffic in China [46] which indicated a positive correlation between the frequency of signaling-intensive traffic and malicious activities such as private data upload and billing fraud. Finally, signaling storms may follow and hence prolong network outages from cyber-attacks, due to the large number of user devices that will attempt to reconnect after the service is restored [19].

2.3. Prior Work on Storm Detection and Mitigation

Online detection of deliberate signaling attacks was first studied in [45], where connection inter-setup times for each mobile are estimated from IP metrics in order to detect the intention of a remote host to launch an attack. A general framework for anomaly detection was presented in [13] based on time-series analysis and change detection algorithms. While the goal of [13,45] is to identify large-scale events by aggregating and analyzing statistics

https://doi.org/10.1017/S0269964816000140 Published online by Cambridge University Press

from all hosts and mobile users, respectively, our approach aims to identify users that are contributing to a problem, namely signaling overload, rather than detect the problem itself. The work in [42] considered the detection of mobile-initiated signaling attacks via a supervised learning approach, which monitors transmissions that trigger a radio access bearer setup procedure, and extracts from the corresponding packets features relating to destination IP and port numbers, packet size and response-request ratio. We utilize similar attributes in our approach, but we do not assume knowledge of the effect that a packet has on the control plane (i.e., whether it has triggered a connection setup procedure), thus simplifying the deployment of our solution in operational networks. In a previous work, [28], we developed a technique which directly monitors the control plane of each active mobile device; it counts the number of successive signaling transitions that do not utilize allocated bandwidth, and temporarily blocks devices that exceed a certain threshold to avoid overloading the network. Although such a signaling-based approach can be more effective in detecting and mitigating storms, it requires changes to network equipment and/or protocols [19,49] which can be slow and costly to implement.

A number of commercial solutions also started to appear in response to recent incidents of signaling storms, which can be classified into three groups. First, anomaly detection and mitigation systems [19] such as [28] and the one presented in this paper. Second, air interface optimization solutions which aim to increase the number of simultaneously connected devices in the access network. Such solutions are constantly evolving with new standards, specifications and proprietary admission/congestion control and scheduling algorithms added all the time; our approach operates on top of and is complimentary to such air interface technologies. Third, dedicated signaling infrastructures to handle the expected growth in CN signaling due to policies, charging, mobility management and other new services offered for the first time in LTE networks. However, it is expected that routing, congestion management and load balancing in the CN will be less of an issue, with the trend towards network functions virtualization that will enable dynamic resource scaling as required by network load.

3. THE RNN

The RNN is a biologically inspired computational model, introduced by Gelenbe [23], in which neurons exchange signals in the form of spikes of unit amplitude. In RNN, positive and negative signals represent excitation and inhibition respectively, and are accumulated in neurons. Positive signals are canceled by negative signals, and neurons may fire if their potential is positive. A signal may leave neuron i for neuron j as a positive signal with probability p^+_{ij}, as a negative signal with probability p^-_{ij}, or may depart from the network with probability d_i, where $\sum_j [p^+_{ij} + p^-_{ij}] + d_i = 1$. Thus, when neuron i is excited, it fires excitatory and inhibitory signals to neuron j with rates:

$$w^+_{ij} = r_i p^+_{ij} \geq 0, \quad w^-_{ij} = r_i p^-_{ij} \geq 0,$$

where

$$r_i = (1 - d_i)^{-1} \sum_j [w^+_{ij} + w^-_{ij}].$$

The steady-state probability that neuron i is excited is given by:

$$q_i = \frac{\Lambda_i + \sum_j q_j w^+_{ji}}{\lambda_i + r_i + \sum_j q_j w^-_{ji}},$$

https://doi.org/10.1017/S0269964816000140 Published online by Cambridge University Press

where Λ_i and λ_i denote the rates of exogenous excitatory and inhibitory signal inputs into neuron i, respectively.

A gradient descent supervised learning algorithm for the recurrent RNN has been developed in [25]. For a RNN with n neurons, the learning algorithm estimates the $n \times n$ weight matrices $\mathbf{W}^+ = \{w_{ij}^+\}$ and $\mathbf{W}^= \{w_{ij}^-\}$ from a training set comprising K input-output pairs $(\mathbf{X}, \mathbf{Y})$. The set of successive inputs to the algorithm is $\mathbf{X} = (\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(K)})$, where $\mathbf{x}^{(k)} = (\mathbf{\Lambda}^{(k)}, \boldsymbol{\lambda}^{(k)})$ are the pairs of exogenous excitatory and inhibitory signals entering each neuron from outside the network:

$$\mathbf{\Lambda}^{(k)} = (\Lambda_1^{(k)}, \ldots, \Lambda_n^{(k)}), \quad \boldsymbol{\lambda}^{(k)} = (\lambda_1^{(k)}, \ldots, \lambda_n^{(k)}).$$

The successive desired outputs are $\mathbf{Y} = (\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(K)})$, where the kth vector $\mathbf{y}^{(k)} = (y_1^{(k)}, \ldots, y_n^{(k)})$, whose elements $y_i^{(k)} \in [0, 1]$ correspond to the desired output values for each neuron. The training examples are presented to the network sequentially, and the weights are updated according to the gradient descent rule to minimize an error function:

$$E^{(k)} = \frac{1}{2} \sum_{i=1}^{n} a_i [q_i^{(k)} - y_i^{(k)}]^2, \quad a_i \geq 0.$$

The update procedure requires a matrix inversion operation for each neuron pairs (i, j) and input k which can be done in time complexity $O(n^3)$, or $O(mn^2)$ if m-step relaxation method is used, and $O(n^2)$ for feed-forward networks.

4. THE DETECTION SYSTEM

Figure 2 shows the basic architecture of the packet-switched domain of mobile networks, which consists of the following elements: the UEs; the RAN which is connected to the CN through a backhaul network; the mobile gateway which allows packets to be exchanged with external networks such as the Internet; and the operation and support system (OSS) which provides network management functions such as monitoring, configuration, service provisioning, etc.

Also shown in Figure 2 is the positioning of the proposed detection system within the mobile network, which intercepts packets directed to/from the network gateway; in

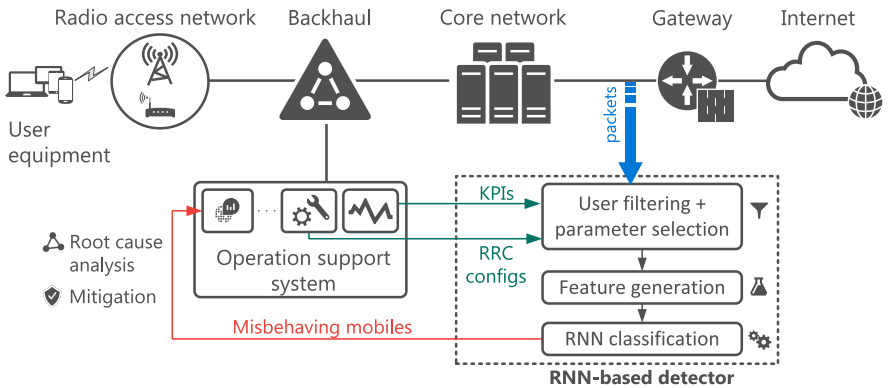

FIGURE 2. The detection system and its interactions with the components of a mobile network.

https://doi.org/10.1017/S0269964816000140 Published online by Cambridge University Press

3GPP standards, the user data transported over this interface are encapsulated in GTP-U (a simple IP-based tunneling protocol) packets. The detector also utilizes information from the OSS to reduce search space and optimize performance, and periodically produces a list of anomalous users to the OSS for root cause analysis and mitigation. The detection algorithm consists of three data processing stages: (i) user filtering and parameter selection based on network configuration settings and key performance indicators (KPIs) related to signaling load on various network components, (ii) feature generation and (iii) user classification with a trained RNN model. For reasons that should become apparent, we describe these different data processing steps in a logical order rather than the order in which they happen during run time.

4.1. Online RNN Classification

The RNN-based algorithm monitors the activity of a set of mobile devices, specified by the data filter, and calculates expressive features that describe various characteristics of the users' behavior. Time is divided into slots, each of duration Δ seconds, in which summary statistics of several quantities related to the IP traffic of each user are collected. The algorithm stores the most recent w set of measurements, and uses them to compute the current values of the input features, that is, the features for time slot τ are computed from measurements obtained for time slots $\tau, \tau-1, \ldots, \tau-(w-1)$ so that the observation window of the algorithm is $W = w\Delta$. Let $z^{(\tau)}$ denote a measured or calculated quantity for time slot τ, then the ith input feature $x_i^{(\tau)}$ is obtained by applying a statistical function ϕ_i of the following form:

$$x_i^{(\tau)} = \phi_i\big(z^{(\tau)}, z^{(\tau-1)}, \ldots, z^{(\tau-w-1)}\big).$$

Hence, by employing different operators ϕ_i on different statistics z stored over the observation window of w slots, it is possible to capture both instantaneous (i.e., sudden) and long-term changes in the traffic profile of a user. In our experiments, we have applied a number of simple statistical functions including:

- The mean and standard deviation of z across the entire window.
- An exponential moving average filter in which the current feature is computed as $x_i^{(\tau)} = \alpha x_i^{(\tau-1)} + (1-\alpha)z^{(\tau)}$, where α is some constant $0 < \alpha < 1$ typically close to 1, with higher values discounting older observations faster.
- Shannon entropy which measures the uncertainty or unpredictability in the data; it is defined as $x_i^{(\tau)} = -\sum_{t=\tau-w-1}^{\tau} p_{z^{(t)}} \log p_{z^{(t)}}$, where $p_{z^{(t)}}$ is the probability of observing data item $z^{(t)}$ within the window, which can be estimated from the histogram of the data. Entropy is typically interpreted as the minimum number of bits required to encode the classification of a data item, thus a small entropy indicates deterministic behavior which is often associated with signaling anomalies [22,55].
- Anomaly score based on how close the measured quantities are to a range of values considered to be suspicious.

Once the input features for a slot have been computed, they are fused using a trained feed-forward RNN architecture such as the one presented in Figure 3 to yield the final decision. The input neurons receive the features computed for the current time slot as exogenous excitatory signals, while all exogenous inhibitory signals are set to zero. The output neurons correspond to the probabilities of the input pattern belonging to any of two traffic classes (i.e., attack or normal). The final decision about the traffic observed in the time slot is determined by the ratio of the two output nodes, which is q_{14}/q_{15} in the figure:

https://doi.org/10.1017/S0269964816000140 Published online by Cambridge University Press

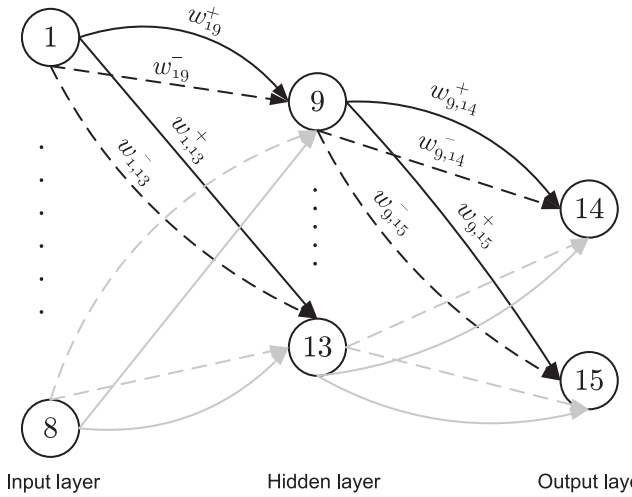

FIGURE **3.** The feed-forward RNN structure used for anomaly detection, with eight input nodes, five hidden neurons and two output nodes corresponding to attack and normal traffic. The learning algorithm processes the input training patterns in sequence and updates the weights. The kth training set consists of a feature vector $\mathbf{x}^{(k)} = (x_1^{(k)}, \ldots, x_8^{(k)})$ comprising exogenous excitatory signals $(\Lambda_1^{(k)}, \ldots, \Lambda_8^{(k)})$, and its classification $\mathbf{y}^{(k)} = (y_{14}^{(k)}, y_{15}^{(k)})$ which is set to $(1, \epsilon)$ for attack and $(\epsilon, 1)$ for normal samples where $\epsilon \simeq 0$. All other exogenous signals are set to zero.

it is classified as attack if the ratio is greater than 1 and normal otherwise. We have used an implementation of the RNN provided in [2].

4.2. Feature Selection

Selecting highly informative features for any classification problem is one of the most important parts of the solution. The features that we wish to use should capture the RRC signaling dynamics of users, be easy to measure or calculate without high computational or storage cost, and reflect both the instantaneous behavior and the long term trend of the traffic. For each mobile under observation, we extract information related to inter-arrival times, lengths and destination IP addresses of packets, which have been suggested previously [13,42,56] as good indicators of signaling misbehavior. The specific features that we have used are described below.

4.2.1. Inter-arrival Time. RRC signaling occurs whenever the UE sends or receives packets following an inactivity period that exceeds an RRC timer. Thus, the volume of traffic exchanged by a UE does not map directly into signaling load which is more influenced by the frequency of intermittent transmissions. To capture this coupling between the data and RRC signaling planes, we define a *burst* as a collection of packets whose inter-arrival times are less than δ seconds, where δ is smaller than the RRC timers, typically in the order of few seconds. Thus, for a sequence of packets whose arrival instants are $\{t_1, t_2, \ldots\}$, we group all packets up to the nth arrival into a single burst, where $n = \inf\{i : t_i - t_{i-1} > \delta\}$, and then proceed in a similar manner starting from the $(n+1)$th packet arrival. Note that a burst may not necessarily generate signaling, even if it arrives after the time-out,

https://doi.org/10.1017/S0269964816000140 Published online by Cambridge University Press

due to potential network delays that may modify inter-arrival times of packets. However, packets within a single burst are likely not to trigger any control plane messages, while inter-arrival times of bursts will be correlated with the actual signaling load generated by the UE. In this manner, we remove any bias regarding the volume of traffic sent or received by the UE, and put more emphasis on the frequency of potentially resource-inefficient communications.

The features based on the times between bursts are then calculated as follows. The algorithm stores the mean and standard deviation of the inter-burst times in each slot then, using the most recent w values, it computes (i) entropy of these average values, (ii) moving average of the standard deviations and (iii) moving average of an anomaly score for the average values computed based on the RRC timer T in the high bandwidth state. In particular, the anomaly score $a(z^{(t)})$ of the average inter-burst time in slot t is set to zero when $z^{(t)} < T$, reflecting the fact that such shortly spaced bursts may not have generated many RRC transitions; it is high when $z^{(t)}$ is slightly larger than T, indicating potentially resource-inefficient bursts; and it drops quickly when $z^{(t)}$ is few seconds larger than T. We obtain this effect using for example a Pareto or gamma density functions that assert $z^{(t)}$ must be greater than $T - \epsilon$, but not too much greater, which can be controlled by adjusting the parameters of the density function.

4.2.2. Packet Size. The packet size distribution for a normal device can be markedly different from that of a device that runs a misbehaving application. For example, when signaling storms occur due to unexpected events in the Internet such as cloud outages [13,55], the client application attempts to reconnect to its servers more frequently, causing significant increase in the number of TCP SYN packets sent by the user. This in turn changes the randomness associated with the size of packets, and can be used to identify misbehaving mobiles in the event of a storm. Our algorithm computes the average size of packets sent by a UE within each slot, and evaluates a feature based on the entropy of the most recent w measurements.

4.2.3. Burst Rate. Another obvious characteristic of signaling storms is the sudden sustained rate acceleration of potentially harmful bursts generated by a misbehaving user. Moving average of the burst rate per slot and entropy of the rates across the observation window are used as features in order to capture, respectively, the frequent and repetitive nature of nuisance transmissions. Furthermore, a misbehaving application may change the traffic profile of a user in terms of the ratio of received and sent bursts (known as response ratio), as in the case of the outage induced storm described above where many SYN packets will not generate acknowledgments. Hence, we also use as a feature the mean of the response ratios within the window of w slots.

4.2.4. Destination Address. The number of destination IP addresses for a normally functioning mobile device can be very different from that of an attacker [42], whether the attack originates from the mobile network due to a misbehaving application, or from the Internet as in the case of unwanted traffic reaching the mobile network [56]. In the former, the number of destination IP addresses will be very small *compared to* the frequency of bursts, while in the latter this number is high. Thus, we calculate the percentage of *unique* destination IP addresses contacted within each time slot, and use the average of the most recent w values as a feature.

https://doi.org/10.1017/S0269964816000140 Published online by Cambridge University Press

4.3. User Filtering and Parameter Selection

Information about the "health" of network servers is typically available to mobile operators in the form of KPIs, which can be fed to the algorithm to determine the users that should be monitored (e.g., those attached to signaling overloaded parts of a network). Also, using KPIs the detector can be switched off when signaling loads are below a certain threshold, effectively eliminating the need to continuously analyze users' traffic. In the following, we summarize the parameters of the RNN algorithm and discuss how they should be selected adaptively, based on both KPIs and RRC configurations, and also how the choice of each parameter influences the performance of the detector:

- Slot size Δ: This defines the resolution of the algorithm and the frequency at which classification decisions are made. It should be long enough for the measured statistical information to be significant, but not too long to make the algorithm react slowly to attacks. In our experiments we set $\Delta = 1\,\text{min}$.
- Window size $W = w\Delta$: This determines the amount of historical information to be included in a classification decision. The choice of the window size presents a trade-off between speed of detection and false alarm rate, since a small window makes the algorithm more sensitive to sudden changes in the traffic profile of a user, which in turn increases both detection and false alarm rates. This trade-off can be optimized by adjusting W according to the level of congestion in the control plane, with shorter windows for higher signaling loads to enable the algorithm to quickly identify misbehaving UEs. The value of w used in our experimental results is 5, but we also experimented with other values which confirmed the aforementioned observations.
- Maximum packet inter-arrival time within a burst δ: This should be selected based on the RRC timers, so that potentially resource-inefficient transmissions can be tracked. In our simulations of a UMTS network, the timers in cell_DCH and cell_FACH states are set to, respectively, $T_1 = 6s$ and $T_2 = 12s$ based on [53]. We have evaluated different values of δ in $0.5\,\text{min}(T_1, T_2) < \delta < \text{min}(T_1, T_2)$, which all led to similar detection performance, but training time tends to drop as δ is increased within this range. This is because the time between malicious transmissions in our training dataset is slightly greater than $\text{min}(T_1, T_2)$, so that the difference between attack and normal traffic becomes more pronounced in the features as δ approaches this value, leading to shorter training times. However, while large values of δ will reduce computational overhead, they may result in the loss of valuable information due to traffic aggregation.

5. EXPERIMENTS AND RESULTS

In this section, we evaluate the performance of our detection technique using the mobile network simulator developed in [39,40]. We first present the traffic models that characterize the normal user behavior, and two attack models that represent both malicious and misbehaving UEs. Then we discuss the results of applying the algorithm on the dataset produced by the simulator.

Since the impact of signaling storms on mobile networks has been analyzed extensively in [4,21,40], the objective of the present simulation setup is to evaluate the performance of our detection algorithm, and therefore only a small scenario has been considered. In particular, we simulated 200 3G/UMTS UEs in an area of $2 \times 2\,\text{km}^2$ which is covered by 7

https://doi.org/10.1017/S0269964816000140 Published online by Cambridge University Press

base stations connected to a single radio network controller (RNC). The CN consists of the SGSN and the GGSN (the mobile gateway) which is connected to 37 Internet hosts acting as application servers, five of which for instant messaging (IM), and two are contacted by the attacking UEs.

5.1. Model of the User

The user model consists of three popular mobile services that are active simultaneously in order to create realistic traffic profiles. The model can also support a diurnal pattern for UE behavior, where the UE is active for a certain duration every 24 h, and is inactive the rest of the time during which the user does not generate or respond to traffic. This pattern represents the day/night cycle of users, and can be varied from one user to another based on a random distribution.

5.1.1. Web Browsing. The interactive web browsing behavior is based on the self-similar traffic model described in [40] and assumes Zipf-like distribution for web server popularity, which has been widely used in the literature since it was first suggested in [11].

5.1.2. Instant Messaging. IM applications are characterized by frequent, small data transmissions and a long tail distribution representing messages with media rich contents such as videos and photos. The IM application model consists of two distinct but related parts: message generator and responder. Each UE generates messages to chosen destinations, and also responds to received messages with a given probability. The message generator works based on sessions and waves. A session represents the duration that the user is actively generating messages, and consists of one or more waves where the messages are actually sent. At each wave, the user generates and sends one or more messages, the number and length of which are configurable with random distributions, to a single destination (mobile user) chosen at random. The time between waves within a session, the session duration and the time between user sessions are all given by random distributions. On the other hand, the UE responds to each received message with a given probability, and this response behavior is independent of message generation, and can occur both inside and outside of the user's IM sessions.

The final destination of a message can be another mobile in the same network (explicitly simulated) or a mobile in another network; mobiles in different networks are represented by one or more servers in the simulation, which act on behalf of these users. Regardless of its final destination, each message passes through an Internet chat server, which forwards the message to its final destination, that is, another mobile user. We simulate multiple chat servers representing popular chat applications and services such as WhatsApp, Skype, etc., and currently assume that each message belongs to a chat application that is chosen uniformly at random from the available applications. The simulation model supports more generic message-to-application assignment based on other random distributions.

5.1.3. Short Message Service. The SMS application operates in the same manner as the IM application, but differs in that there is a single intermediate server within the mobile network that handles all SMS messages for that network, that is, the SMSC server. SMS messages are also different than IM messages in their types, which can be in-network mobile, out-network mobile, premium, etc. In-network mobiles are naturally represented by the UEs explicitly simulated, while all other destinations are represented by servers outside

https://doi.org/10.1017/S0269964816000140 Published online by Cambridge University Press

the simulated mobile network, with one or more servers representing each class. Therefore, the type of a sent or received SMS can be inferred from its source and destination addresses (numbers). The type of the SMS message the UE generates is chosen at random based on the parameters of the SMS application. Note that while SMS traffic affects the signaling behavior of users, it is not monitored by our detection system.

5.2. Attack Model

We consider two types of *cell_DCH attacks* which overload the control plane by causing superfluous promotions to the high bandwidth cell_DCH state. The first attack is *aggressive* in the sense that a malicious device knows when an RRC state transition occurs, and launches the next attack once a demotion from high to low bandwidth states is detected. To perform the attack, we assume that the attacker has inferred the values of the RRC timers, and is monitoring the user's activity in order to estimate when a transition occurs so as to trigger a new one immediately afterwards. However, there could be an error between the actual transition time and the estimated one, which we represent by an exponentially distributed random variable with mean $2s$. When the attacker "thinks" that a transition has occurred, it sends a high data rate traffic to one of its Internet servers in order to cause a promotion to the high bandwidth state. This model is used mainly for *training* the RNN.

The second attack type is based on a *poorly designed* mobile application or operating system that sends periodic messages whenever the user is inactive, with the transmission period set to be slightly larger than the cell_DCH timer in order to increase the chances of triggering state transitions. This behavior represents, for example, the case where a pull mechanism is used to fetch updates periodically, and the update period happens to "synchronize" with the RRC timer. However, unlike aggressive attackers, this behavior does not guarantee the generation of signaling traffic for each data transfer, since (i) it only starts when local user activity stops but there can be downlink traffic that may have restarted the timer at the signaling server; and (ii) the data volume may not be large enough to trigger a promotion to cell_DCH state. In both cases, the periodic transmissions may become completely *out of sync* with the RRC state machine, therefore not generating significant signaling traffic.

The two distinct attack models allow us to represent both malicious and benign behaviors that may lead to a storm, but the first is well distinguishable and separable from the behavior of a normal user, in terms of both temporal and traffic volume characteristics. On the other hand, the second attack model captures the signaling behavior of legitimate applications and operating systems that are much more similar to an "attack" rather than to a "normal" behavior, but are difficult to detect from user plane dynamics. Thus we use this model to *test* the performance of our algorithm.

5.3. Results

The RNN algorithm provides at the end of a time slot the probabilities that the input features belong to an attack and normal behavior, and the final decision about the traffic is then determined by the ratio of the two output nodes: It is classified as attack if the ratio is greater than 1 and normal otherwise. Figure 4 shows the classifier output (top) and the actual RRC state transitions (bottom) of a *misbehaving* UE as captured during a simulation run. It can be observed that when the malfunctioning application is active, the number of state transitions significantly increases, with most transitions occurring between the cell_FACH and cell_DCH states in this attack scenario. This alternating behavior causes

https://doi.org/10.1017/S0269964816000140 Published online by Cambridge University Press

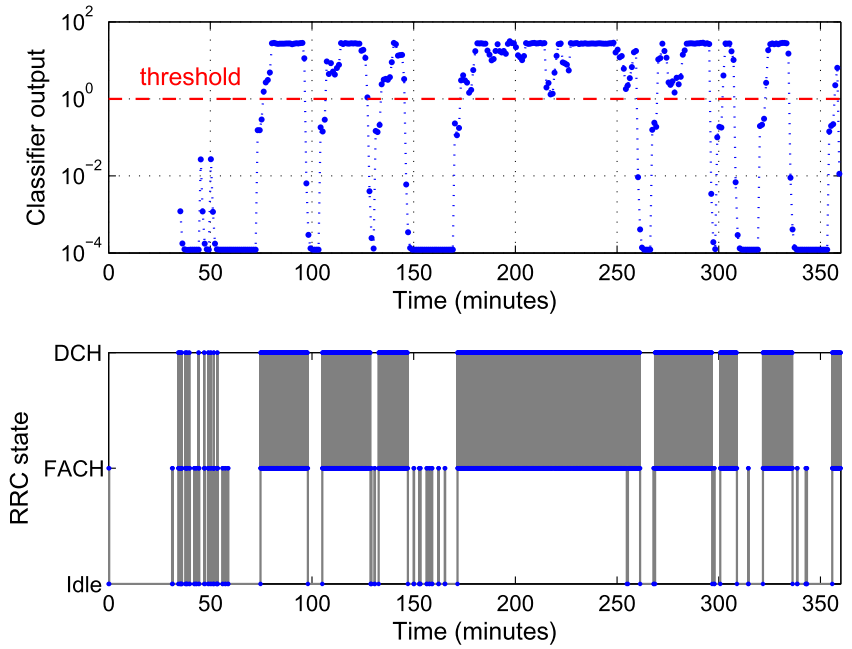

FIGURE 4. Classifier output (top) and state transitions (bottom) for a misbehaving UE.

excessive signaling load in the mobile network, while predominantly generating normal traffic volume, rendering traditional DoS defense techniques ineffective. However, our detection mechanism is able to track very accurately the RRC state transitions of the UE, and to identify quickly when excessive signaling is being generated, despite the fact that it does not directly monitor these transitions but rather infers them from the features that we have described. One can also observe that the classifier's output sometimes drops close to 1 during an attack epoch, which is attributed to other normal applications generating traffic in those time instants, thus reducing the severity of the attack. As mentioned earlier, the detection speed and tolerance to signaling misbehavior can be adjusted by modifying the size of the observation window, which in this scenario is set to 5 min.

Figure 5 shows results when there is no attack, where the number of state transitions in a given period are small and due to normal traffic generated and received by the UE. In this case, the classifier does not generate any alarms regarding the signaling behavior of the UE as one would expect.

Next we examine in Figure 6 how our algorithm performs when presented with a normal user that generates moderately more state transitions than the average normal user in the simulations. Interestingly enough, the classifier outputs a single alarm (out of 360 samples) when the corresponding state transitions are indeed excessive. Since the detection algorithm is supposed to be active only when there is a signaling overload condition, such classification decisions may not always be considered as false alarms, as the goal would be to identify users that are causing congestion, regardless of whether they are attacking deliberately or not.

Finally, Figure 7 illustrates the accuracy of our classifier, namely the proportion of correct decisions (both true positives and true negatives) out of all test samples. The figure shows results for 50 UEs, where each data point represents the average of 360 classification decisions taken during the simulation experiment which lasted for 6 hours (note

https://doi.org/10.1017/S0269964816000140 Published online by Cambridge University Press

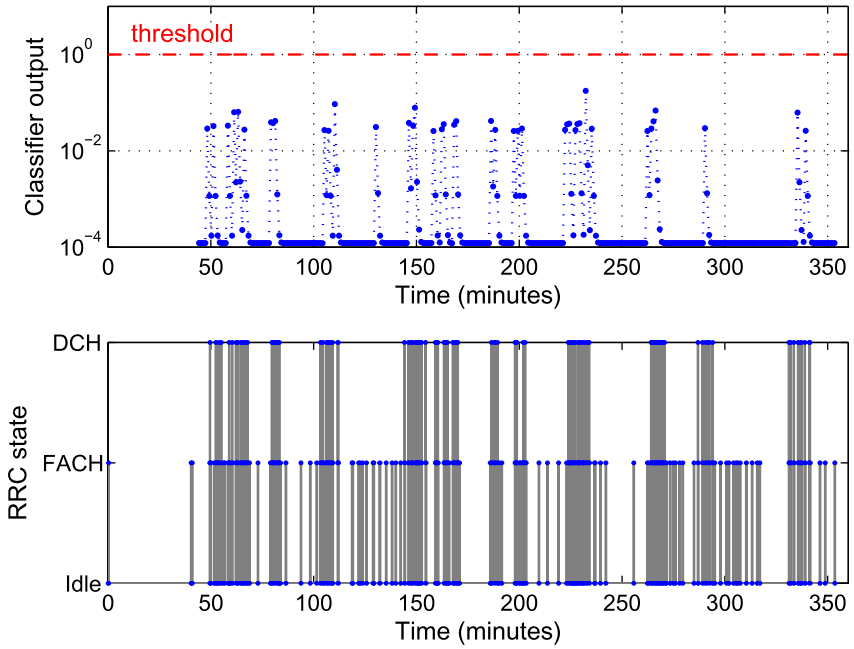

FIGURE **5.** Classifier output (top) and RRC state transitions (bottom) for a normal UE.

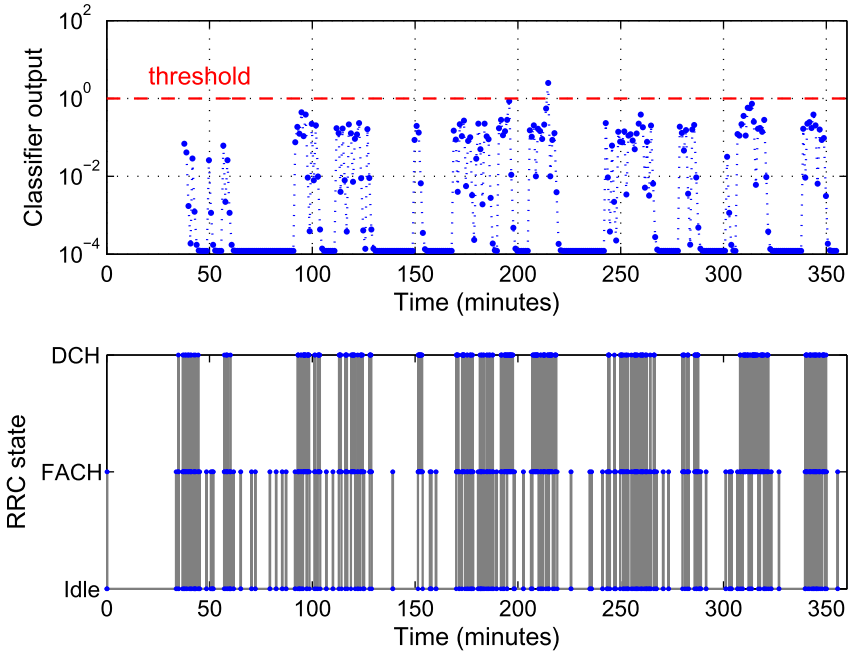

FIGURE **6.** Classifier output (top) and state transitions (bottom) for a heavy normal UE.

the resolution of the detector is $\Delta = 1\,\text{min}$). The evaluation is based on a strict criterion whereby for each UE, we assume that if it generates at least 1 attack packet within a time slot, then the corresponding output of the classifier should be greater than 1, otherwise a

https://doi.org/10.1017/S0269964816000140 Published online by Cambridge University Press

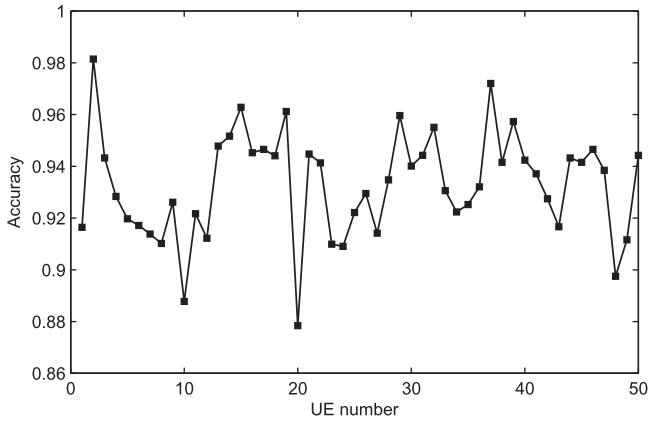

FIGURE **7.** The accuracy of the RNN algorithm, measured as the fraction of correct decisions over the activity period of 6 h, for 50 misbehaving UEs.

false classification decision is declared. The results indicate an accuracy between 88% and 98% with an average of 93% over the 50 test cases. This fluctuation can be attributed to the fact that our algorithm does not classify an attack as such until few time slots have passed (depending on the number of slots w within the window), and therefore misbehaving UEs with many silent periods will produce higher false positives; fortunately, these less aggressive UEs will generate lower signaling load. In fact, the fraction of normal instances that have been mistakenly classified as attack (false positive rate) is zero for all but one case, while the fraction of attack instances that have been correctly identified (true positive rate) is on average 90% which can be improved, at the cost of higher false positives, by reducing the window size W.

6. CONCLUSIONS

This paper proposed an online approach for detecting mobile devices that contribute to signaling overload, based on the RNN [23,25]. The method relies on the analysis of data packets traversing the mobile CN, which can be performed using standard traffic monitoring tools, in order to infer the signaling dynamics of users. This offers the advantages of not requiring to decode/decrypt lower control plane layers, and fewer number of nodes to monitor. In the algorithm, summary statistics about the behavior of each observed mobile device are collected and stored in a moving window at fixed time intervals (slots), from which a number of features are calculated to capture both sudden and long term changes in the user's signaling behavior. The features for the most recent time slot are subsequently fused using a trained RNN to produce the final classification decision. Using a discrete-event mobile network simulator, we have shown that our technique achieves a very high detection rate with almost zero false alarms. The proposed approach is also flexible, providing a number of parameters to optimize the trade-off between detection speed, accuracy and overhead, based on signaling overload conditions in the network.

Acknowledgement

Thanks to Gokce Gorbil for providing the simulation data. This work was partially supported by the EU FP7 project NEMESYS (Enhanced Network Security for Seamless Service Provisioning in the Smart Mobile Ecosystem), under grant agreement no. 317888.

References

1. 3GPP TR 23.887: Machine-type and other mobile data applications communications enhancements (release 12). http://www.3gpp.org/DynaReport/23887.htm, 3GPP, December 2013, Technical Report.
2. Abdelbaki, H. Matlab simulator for the RNN. [Online]. Available: http://www/cs/ucf.edu/ahossam/rnnsim
3. Abdelrahman, O.H. & Gelenbe, E. (2012). Packet delay and energy consumption in non-homogeneous networks. *Computer Journal* 55(8): 950–964.
4. Abdelrahman, O.H. & Gelenbe, E. (2014). Signalling storms in 3G mobile networks. In *Proceedings of IEEE International Conference on Communications (ICC)*, Sydney, Australia, June 2014, pp. 1017–1022.
5. Abdelrahman, O.H., Gelenbe, E., Gorbil, G., & Oklander, B. (2013). Mobile network anomaly detection and mitigation: the NEMESYS approach. In *Proceedings of 28th International Symposium on Computer and Information Sciences (ISCIS)*, ser. LNEE. Springer, Paris, France, October 2013, vol. 264, pp. 429–438.
6. Amrutkar, C. *et al.* (2013). Why is my smartphone slow? On the fly diagnosis of underperformance on the mobile internet. In *Proceedings of 43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. Budapest, Hungary: IEEE Computer Society, June 2013, pp. 1–8.
7. Arbor Networks. (2014). Worldwide infrastructure security report. 2014. [Online]. Available: http://pages.arbornetworks.com/rs/arbor/images/WISR2014.pdf
8. AT&T. (2012). Best practices for 3G and 4G app development. Whitepaper, 2012. [Online]. Available: http://developer.att.com/static-assets/documents/library/best-practices-3g-4g-app-development.pdf
9. Atalay, V. & Gelenbe, E. (1992). Parallel algorithm for colour texture generation using the random neural network model. *International Journal of Pattern Recognition and Artificial Intelligence* 6(02–03): 437–446.
10. Atalay, V., Gelenbe, E., & Yalabik, N. (1992). The random neural network model for texture generation. *International Journal of Pattern Recognition and Artificial Intelligence* 6(1): 131–141.
11. Breslau, L., Cao, P., Fan, L., Phillips, G., & Shenker, S. (1999). Web caching and Zipf-like distributions: evidence and implications. In *Proceedings of IEEE INFOCOM*, New York, NY, USA, vol. 1, March 1999, pp. 126–134.
12. Choi, Y., Yoon, C., Kim, Y., Heo, S.W., & Silvester, J. (2014). The impact of application signaling traffic on public land mobile networks. *IEEE Communications Magazine* 52(1): 166–172.
13. Coluccia, A., D'alconzo, A. & Ricciato, F. (2013). Distribution-based anomaly detection via generalized likelihood ratio test: A general maximum entropy approach. *Comput. Netw.* 57(17): 3446–3462.
14. Corner, S. (2011). Angry birds + android + ads = network overload. June 2011. [Online]. Available: http://www.itwire.com/business-it-news/networking/47823
15. Cramer, C., Gelenbe, E., & Bakircloglu, H. (1996). Low bit-rate video compression with neural networks and temporal subsampling. *Proceedings of the IEEE* 84(10): 1529–1543.
16. Cramer, C. & Gelenbe, E. (2000). Video quality and traffic QoS in learning-based subsampled and receiver-interpolated video sequences. *IEEE Journal on Selected Areas in Communications* 18(2): 150–167.
17. Cramer, C., Gelenbe, E., & Gelenbe, P. (1998). Image and video compression. *IEEE Potentials* 17(1): 29–33.
18. Donegan, M. (2011). Operators urge action against chatty apps. Light Reading Report, June 2011. [Online]. Available: http://www.lightreading.com/operators-urge-action-against-chatty-apps/d/d-id/687399
19. Ericsson. (2014). High availability is more than five nines. July 2014. [Online]. Available: http://www.ericsson.com/real-performance/wp-content/uploads/sites/3/2014/07/high-avaialbility.pdf
20. Ericsson. (2014). A smartphone app developer's guide: Optimizing for mobile networks. Whitepaper, April 2014. [Online]. Available: http://www.ericsson.com/res/docs/2014/smartphone-app-dev-guide.pdf
21. Francois, F., Abdelrahman, O.H., & Gelenbe, E. (2015). Impact of signaling storms on energy consumption and latency of LTE user equipment. In *Proceedings of Seventh IEEE International Symposium on Cyberspace Safety and Security (CSS)*, New York, August 2015.
22. Gabriel, C. (2012). DoCoMo demands Google's help with signalling storm. Rethink Wireless, January 2012. [Online]. Available: http://www.rethink-wireless.com/2012/01/30/docomo-demands-googles-signalling-storm.htm
23. Gelenbe, E. (1989). Random neural networks with negative and positive signals and product form solution. *Neural Computation* 1(4): 502–510, [Online]. Available: http://dx.doi.org/10.1162/neco.1989.1.4.502
24. Gelenbe, E. (1990). Stability of the random neural network model. *Neural Computation* 2(2): 239–247.

25. Gelenbe, E. (1993). Learning in the recurrent random neural network. *Neural Computation* 5(1): 154–164. [Online]. Available: http://dx.doi.org/10.1162/neco.1993.5.1.154
26. Gelenbe, E. (1994). G-networks: a unifying model for neural and queueing networks. *Annals of Operations Research* 48(5): 433–461. [Online]. Available: http://dx.doi.org/10.1007/BF02033314
27. Gelenbe, E. (2000). The first decade of G-networks. *European Journal of Operational Research* 126(2): 231–232. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377221799004750
28. Gelenbe, E. & Abdelrahman, O.H. (2015). Countering mobile signaling storms with counters. In *Proceedings of International Conference on Cyber PhysicaL Systems, IoT and Sensors Networks (Cyclone)*, Rome, Italy, October 2015.
29. Gelenbe, E., Bakircioglu, H., & Kocak, T. (1998). Image processing with the random neural network. pp. 38–49. [Online]. Available: http://dx.doi.org/10.1117/12.304658
30. Gelenbe, E., Feng, Y., & Krishnan, K. (1996). Neural network methods for volumetric magnetic resonance imaging of the human brain. *Proceedings of the IEEE* 84(10): 1488–1496.
31. Gelenbe, E. & Fourneau, J.-M. (1999). Random neural networks with multiple classes of signals. *Neural Computation* 11(4): 953–963, [Online]. Available: http://dx.doi.org/10.1162/089976699300016520
32. Gelenbe, E., Harmani, K., & Krolik, J. (1998). Learning neural networks for detection and classification of synchronous recurrent transient signals. *Signal Processing* 64(3): 233–247, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S016516849700193X
33. Gelenbe, E. & Hussain, K. (2002). Learning in the multiple class random neural network. *IEEE Transactions on Neural Networks* 13(6): 1257–1267.
34. Gelenbe, E., Koçak, T., & Wang, R. (2004). Wafer surface reconstruction from top-down scanning electron microscope images. *Microelectronic Engineering* 75(2): 216–233, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167931704003302
35. Gelenbe, E. & Loukas, G. (2007). A self-aware approach to denial of service defence. *Computer Networks* 51(5): 1299–1314.
36. Gelenbe, E. & Morfopoulou, C. (2011). A framework for energy-aware routing in packet networks. *Computer Journal* 54(6): 850–859.
37. Gelenbe, E., Sungur, M., Cramer, C., & Gelenbe, P. (1996). Traffic and video quality with adaptive neural compression. *Multimedia Systems* 4(6): 357–369, [Online]. Available: http://dx.doi.org/10.1007/s005300050037
38. Gelenbe, E. & Timotheou, S. (2008). Random neural networks with synchronized interactions. *Neural Computation* 20(9): 2308–2324, [Online]. Available: http://dx.doi.org/10.1162/neco.2008.04-07-509
39. Gorbil, G., Abdelrahman, O.H., & Gelenbe, E. (2014). Storms in mobile networks. In *Proceedings of Tenth ACM Symposium on QoS and Security for Wireless and Mobile Networks (Q2SWinet)*, Montreal, Canada, September 2014, pp. 119–126.
40. Gorbil, G., Abdelrahman, O.H., Pavloski, M., & Gelenbe, E. (2015). Modeling and analysis of RRC-based signalling storms in 3G networks. *IEEE Transactions on Emerging Topics in Computing* PP(99): 1, [Online]. Available: http://dx.doi.org/10.1109/TETC.2015.2389662
41. GSMA. (2014). Smarter apps for smarter phones, version 4.0. November 2014. [Online]. Available: http://www.gsma.com/newsroom/wp-content/uploads//TS-20-v4-0.pdf
42. Gupta, A., Verma, T., Bali, S., & Kaul, S. (2013). Detecting MS initiated signaling DDoS attacks in 3G/4G wireless networks. In *Proceedings of Fifth International Conference on Communication Systems and Networks (COMSNETS)*, Bangalore, India, January 2013, pp. 1–6.
43. Jiantao, S. (2012). Analyzing the network friendliness of mobile applications. Huawei, Technical Report, July 2012. [Online]. Available: http://www.huawei.com/ilink/en/download/HW_146595
44. Ksentini, A., Hadjadj-Aoul, Y., & Taleb, T. (2012). Cellular-based machine-to-machine: overload control. *IEEE Network* 26(6): pp. 54–60.
45. Lee, P.P., Bu, T., & Woo, T. (2007). On the detection of signaling DoS attacks on 3G wireless networks. In *Proceedings of 26th IEEE International Conference on Computer Communications (INFOCOM)*, Anchorage, AK, USA, May 2007, pp. 1289–1297.
46. Li, J., Pei, W., & Cao, Z. (2013). Characterizing high-frequency subscriber sessions in cellular data networks. In *Proceedings of IFIP Networking Conference*, Brooklyn, NY, May 2013, pp. 1–9.
47. Loukas, G. & Öke, G. (2010). Protection against denial of service attacks: a survey. *Computer Journal* 53(7): 1020–1037.
48. Maslennikov, D. (2013). Mobile malware evolution: Part 6. Kaspersky Laboratory, Technical Report, February 2013. [Online]. Available: https://securelist.com/analysis/publications/36996/mobile-malware-evolution-part-6/
49. Mulliner, C., Liebergeld, S., Lange, M., & Seifert, J.-P. (2012). Taming Mr Hayes: mitigating signaling based attacks on smartphones. In *Proceedings of 42nd Annual IEEE/IFIP Int'l Conference on Dependable Systems and Networks (DSN)*. Boston, MA: IEEE Computer Society, June 2012, pp. 1–12.

50. NSN Smart Labs. (2011). Understanding smartphone behavior in the network. White paper, January 2011. [Online]. Available: http://networks.nokia.com/system/files/document/nsn_smart_labs_white_paper.pdf
51. Öke, G. & Loukas, G. (2007). A denial of service detector based on maximum likelihood detection and the random neural network. *Computer Journal* 50(6): 717–727.
52. Oke, G., Loukas, G., & Gelenbe, E. (2007). Detecting denial of service attacks with bayesian classifiers and the random neural network. In *Proceedings of Fuzzy Systems Conference (Fuzz-IEEE)*, London, UK, July 2007, pp. 1964–1969.
53. Qian, F., Wang, Z., Gerber, A., Mao, Z.M., Sen, S., & Spatscheck, O. (2010). Characterizing radio resource allocation for 3G networks. In *Proceedings of Tenth ACM Internet Measurement Conference (IMC)*, Melbourne, Australia, November 2010, pp. 137–150.
54. Qian, Z., Wang, Z., Xu, Q., Mao, Z.M., Zhang, M., & Wang, Y.-M. (2012). You can run, but you can't hide: Exposing network location for targeted DoS attacks in cellular networks. In *Proceedings of Network and Distributed System Security Symp. (NDSS)*, San Diego, CA, February 2012, pp. 1–16.
55. Redding, G. (2013). OTT service blackouts trigger signaling overload in mobile networks. Nokia Solutions and Networks, September 2013. [Online]. Available: https://blog.networks.nokia.com/mobile-networks/2013/09/16/ott-service-blackouts-trigger-signaling-overload-in-mobile-networks/
56. Ricciato, F. (2006). Unwanted traffic in 3G networks. *ACM SIGCOMM Computer Communications Reviews* 36(2): 53–56.
57. Ricciato, F., Coluccia, A., & D'Alconzo, A. (2010). A review of DoS attack models for 3G cellular networks from a system-design perspective. *Computer Communications* 33(5): 551–558.
58. Serror, J., Zang, H., & Bolot, J.C. (2006). Impact of paging channel overloads or attacks on a cellular network. In *Proceedings of Fifth ACM Workshop on Wireless Security (WiSe'06)*, LA, CA, September 2006, pp. 75–84.
59. Shafiq, M.Z., Ji, L., Liu, A.X., Pang, J., & Wang, J. (2012). A first look at cellular machine-to-machine traffic: Large scale measurement and characterization. *SIGMETRICS Performance Evaluation Review* 40(1): 65–76.
60. Telesoft Technologies. (2012). Mobile data monitoring. White paper, 2012.
61. Timotheou, S. (2010). The random neural network: A survey. *Computer Journal* 53(3): 251–267, [Online]. Available: http://comjnl.oxfordjournals.org/content/53/3/251.abstract
62. Traynor, P. et al. (2009). On cellular botnets: measuring the impact of malicious devices on a cellular network core. In *Proceedings of 16th ACM Conference on Computer and Communications Security (CCS)*, Chicago, IL, pp. 223–234.
63. Wang, Z., Qian, Z., Xu, Q., Mao, Z., & Zhang, M. (2011). An untold story of middleboxes in cellular networks. In *Proceedings of ACM SIGCOMM*, Toronto, Canada, August 2011, pp. 374–385.

https://doi.org/10.1017/S0269964816000140 Published online by Cambridge University Press