How Can the Adversary Effectively Identify Cellular IoT Devices Using LSTM Networks?

Zhengping Jay Luo*
Rider University
Lawrenceville, New Jersey, USA
zluo@rider.edu

Will A. Pitera
Rider University
Lawrenceville, New Jersey, USA
piteraw@rider.edu

Shangqing Zhao
University of Oklahoma
Oklahoma, USA
shangqing@ou.edu

Zhuo Lu
University of South Florida
Tampa, Florida, USA
zhuolu@usf.edu

Yalin E. Sagduyu
Virginia Tech
Virginia, USA
ysagduyu@vt.edu

ABSTRACT

The Internet of Things (IoT) has become a key enabler for connecting edge devices with each other and the internet. Massive IoT services provided by cellular networks offer various applications such as smart metering and smart cities. Security of the massive IoT devices working alongside traditional devices such as smartphones and laptops has become a major concern. Protecting these IoT devices from being identified by malicious attackers is often the first line of defense for cellular IoT devices. In this paper, we provide an effective attacking method for identifying cellular IoT devices from cellular networks. Inspired by the characteristics of Long Short-Term Memory (LSTM) networks, we have developed a method that can not only capture context information but also adapt to the dynamic changes of the environment over time. Experimental validation shows a high detection rate with less than 10 epochs of training on public datasets.

CCS CONCEPTS

• **Security and privacy → Mobile and wireless security**; • **Networks → Network measurement**.

KEYWORDS

Internet of Things; cellular IoT devices; security; attacks; defenses; LSTM networks; recurrent neural networks

ACM Reference Format:
Zhengping Jay Luo, Will A. Pitera, Shangqing Zhao, Zhuo Lu, and Yalin E. Sagduyu. 2023. How Can the Adversary Effectively Identify Cellular IoT Devices Using LSTM Networks?. In *Proceedings of the 2023 ACM Workshop on Wireless Security and Machine Learning (WiseML '23), June 1, 2023, Guildford, United Kingdom.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3586209.3591394

WiseML '23, June 1, 2023, Guildford, United Kingdom

ACM ISBN 979-8-4007-0133-7/23/06…$15.00
https://doi.org/10.1145/3586209.3591394

1 INTRODUCTION

Internet of Things (IoT) gained momentum following the development of wireless communications and smart device technologies. It is capable of connecting billions of objects in our daily life to the Internet, which will benefit the society fundamentally through seamless communication and automated management systems [10]. To support IoT applications, different types of mobile communication technologies and protocols are proposed to provide the required infrastructure. Cellular networks prove the key means to support the growing number of IoT services.

Massive IoT devices that work in conjunction with the traditional cellular devices, e.g., smartphones and laptops, within the cellular networks have unique characteristics. In particular, they are usually low-cost devices, consume much less energy and the data exchange is much less than traditional cellular devices. Consequently, they are also usually much less capable than traditional cellular devices in defending against attackers [18]. From an attacker's perspective, identification of the cellular IoT devices is the first step of making consequential sabotages of the IoT-driven services.

There are different strategies to identify IoT devices within cellular networks. They are: (i) MAC address and DHCP negotiation-based methods [15]; (ii) machine learning-based methods [7, 8, 15] and (iii) TCP ACK packet-based probing methods [18]. However, we find that MAC addresses and DHCP negotiation-based methods are not sufficient, and traditional machine learning-based methods generally require large volume of training data samples with different features. The TCP ACK packet-based probing method proposed in [18] is not effective enough to counter different real-world situations (e.g., due to latency). After data analysis on real-world datasets, we observe that there are two major reasons that will complicate the identification process in TCP ACK packet-based probing method [18]: end-to-end latency over the network and the alternated operation modes of real world IoT devices. We propose an effective and efficient method in this work based on TCP ACK packet-based probing method through employing the Long Short-Term (LSTM) as a recurrent neural network (RNN) model. LSTM is highly effective in processing sequences of data and capturing temporal correlations [14, 16].

The main idea of the attack method is that an attacker will send a special type of TCP ACK probing packets to both IoT and non-IoT devices in a cellular network using a fixed interval pattern and collecting the response ACK packets. We then embed information

of those probes that failed to receive a response from the probed device into feature vectors that we use to train the LSTM model. Different reasons such as network latency, packet loss, sleeping mode of IoT devices may lead to the failure of a response packet for the probing packet.

We use two real-world public datasets [5, 11] to analyze the end-to-end latency of IoT devices and the alternated operation modes of IoT devices. Furthermore, we combine these two datasets together to simulate the complicated real-world environment and validate our proposed method. Our experimental results show a 100% detection accuracy on the test data within less than 10 epochs of training on the training data. We also discuss potential defense methods.

2 BACKGROUND AND PRELIMINARIES

2.1 Cellular IoT networks and devices

Essentially, The IoT consists of networks, devices, and data [17]. These three components are intertwined together to provide services. Networks that can be used by IoT devices include non-terrestrial networks (e.g., UAV and satellite networks), cellular networks (e.g., 2G/3G/4G/5G and beyond mobile networks, LTE-M network), wide area network (e.g., LoRa, Sigfox) and short-range networks (e.g., WiFi, Bluetooth). IoT devices using cellular networks, also known as cellular IoT devices, have been an emerging solution to connect IoT devices from a broad area. In this work, we study the vulnerabilities and security concerns within cellular IoT networks.

The simplified architecture of cellular IoT networks is shown in Fig. 1. Both IoT and non-IoT devices are connected to the core network through Radio Access Network (RAN). Other networks such as Public Switched Telephone Network (PSTN), Centralized/Cloud-based RAN (C-RAN) and Internet are also connected to the core network. In this architecture, both cellular IoT devices and non-IoT devices use the same RAN to connect to the core network. Each of the two types of devices are assigned an IP address for each device. In this work, we do not consider IoT devices using WiFi and their identification [3]. We only consider the security of cellular IoT devices that connect directly to cellular networks.

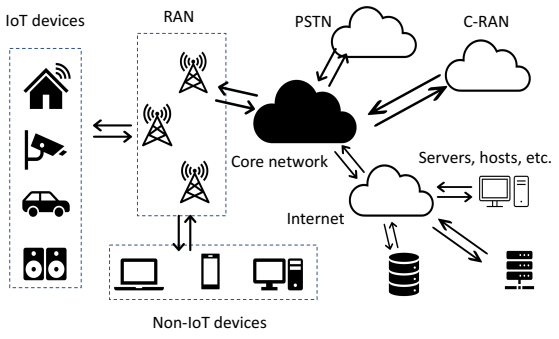

Figure 1: The simplified cellular IoT network architecture.

Based on different applications, IoT can be categorized into two types: massive IoT and critical IoT applications. Massive IoT applications require a large number of low-power devices to collect data in different ways, such as IoT applications used in smart homes and smart cities. Critical IoT is a type of IoT applications that require much fewer devices, in which each device usually handles large volumes of data, e.g., IoT applications used in industrial control and autonomous vehicles. That is why critical IoT application devices usually are more powerful in terms of capability to communicate with the data center and have a strong power supply.

Devices used in massive IoT applications often do not need to transmit large volumes of data, but the number of devices is large. Also, massive IoT devices are expected to work a much longer time with low power supply. Therefore, these IoT devices usually have a Power Saving Mode (PSM) to save energy, in which cellular IoT devices are powered off and not answering any connection requests, but remain still registered in the network. Massive IoT devices can enter *sleeping/inactive* mode when they have no data to transmit. It recovers back to the *active* mode only when they need to transmit data or perform other functional responsibilities. We focus on massive IoT application devices that need to alternate between sleeping and active modes to save power consumption in this work.

2.2 Cellular IoT device identification

With the proliferation of IoT applications, how to identify IoT devices from all the end devices connected to a network becomes a key task [15]. For example, when managing the assets in an organization, system administrators want to know what types of IoT devices are connected and determine if each device is working as expected. IoT devices need to be managed and maintained by the network administrators or the owners of the devices given that IoT devices usually have less complicated structures and capabilities compared to traditional non-IoT devices like phones and laptops.

Obtaining "visibility" of IoT devices in a network is also quintessential for the management of the network, especially from a security perspective. From an attacker's perspective, if the attacker can precisely identify cellular IoT devices from non-IoT devices, then further attacks may be launched such as spamming IoT devices to consume the data plan and further exhaust the network resources [18]. There are various methods to identify IoT devices in a cellular network that also have a lot of non-IoT devices connected to this network. We categorize them into three major directions:

MAC address and DHCP negotiation-based methods An intuitive thinking of identifying IoT devices within a cellular network is that cellular IoT devices and non-IoT devices can be classified using their MAC addresses and DHCP negotiations. However, there are multiple factors that makes this infeasible in real-word scenarios [15]. The MAC address may not convey any information that can be used to identify IoT devices. MAC addresses can be spoofed, and the host name used in DHCP can be changed or not meaningful. Therefore, using MAC addresses and DHCP negotiations are not sufficient to address the problem.

Machine learning-based methods Machine learning models have been widely used in IoT scenarios regarding different applications [6, 9, 12, 19]. To improve the identification performance, a machine learning-based method was proposed in [15] to identify IoT devices based on statistical attributes such as activity cycles, port numbers, signaling patterns and cipher suites. Other methods such as [7, 8] used different modeling strategies to classify IoT

traffic. However, most of the machine learning-based classification methods require a large volume of data that has different features embedded over a long time to train learning models [4]. Nonetheless, these features may not be available for training. For example, the cipher suites are not fixed and various encryption methods can be used in real traffic. A high requirement for the data will make machine learning-based methods difficult to generalize to broader scenarios.

TCP ACK packet-based probing methods A much more simple but efficient cellular IoT identification method was proposed in [18], in which the adversary can interact with the targeted device by sending out a special type of TCP ACK probing packets and collecting the corresponding responses. TCP ACK packets can be used to indicate the types of IoT devices and non-IoT devices. When an ACK packet is received, the recipient needs to confirm the packet by replying another ACK packet. If no response is received by the sender, it implies that this device is either an IoT device in its sleeping mode or an offline non-IoT device. This method relies only on the ACK packets of the end-devices compared to the more complex data requirements of machine learning-based identification methods.

The basic attack procedure of the TCP ACK packet-based probing method starts with the attacker sending out multiple probing packets to an IP address within a network at different time periods to make sure that at least one probing packet is sent out in a time period that the device will fall into a sleeping mode if it is a cellular IoT device. To ensure the reliability of the TCP ACK packet-based probing method, several measures are employed. For example, to ensure that one or more probe packets can fall into the sleeping mode if the device is an IoT device, multiple probes are scheduled with a set of intervals, in which for each possible active time length (each PSM-enabled cellular IoT device has three different active time periods [1] that need to be configured), there exists at least one interval value greater than the active time and smaller than the sum of the active time and the minimum sleeping time. This ensures that at least one of the probes will fall into the sleeping mode. In [18], the probing intervals are set as 15s, 30s, 60s, 180s and 300s.

In this paper, we discuss how an attack may fail when using TCP ACK packet-based method to identify IoT devices and we propose a more powerful mechanism to strengthen the identification accuracy by applying LSTM networks, which is a variant of RNNs.

2.3 Long Short-term Memory (LSTM) networks

RNNs, especially LSTM networks have achieved promising performances on sequential tasks, such as speech recognition, time series prediction, and handwriting recognition [16]. The identification of IoT devices from non-IoT devices within a cellular network based on the interaction of the attacker and devices can be interpreted essentially as a classification problem based on time series data.

The reason why RNNs such as LSTM networks have promising performances on sequential data rather than other deep learning models is largely due to the feedback mechanism in RNNs [14]. It models a dynamic system where the hidden state h_t depend not only on the input of the LSTM layer, but also the previous hidden state h_{t-1}. We call the output of the LSTM layer as hidden state as the LSTM output is usually "hidden" within the input and output of the overall network model. The way an LSTM network works is similar to a deep learning network but replaces the hidden layers in a deep learning model with LSTM layers.

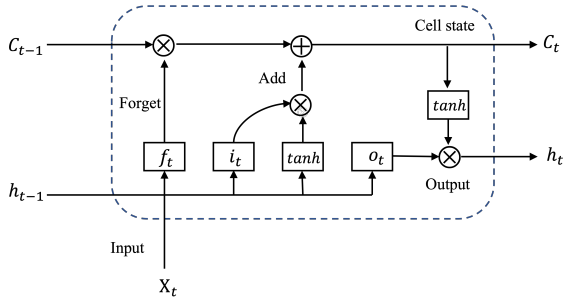

Figure 2: The unit structure of LSTM.

A unit structure of an LSTM is shown in Fig. 2. We can observe from the structure that the core part of a LSTM network model is a memory unit $\mathbf{C}_t$, which is used to store context information. Other gate functions are defined as: $\mathbf{f}_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$, $\mathbf{i}_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$, $\mathbf{g}_t = \sigma(W_g \cdot [h_{t-1}, x_t] + b_g)$, $\mathbf{C}_t = \mathbf{f}_t * \mathbf{C}_{t-1} + \mathbf{i}_t * \mathbf{g}_t$, $\mathbf{o}_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$ and $\mathbf{h}_t = \mathbf{O}_t * \tanh(\mathbf{C}_t)$, in which $\mathbf{f}_t$ denotes the forget gate, $\mathbf{i}_t$ denotes the input gate, $\mathbf{g}_t$ denotes the input modulation gate, $\mathbf{o}_t$ denotes the output gate. σ denotes the sigmoid function that maps the output between 0 and 1. W is the corresponding weight vector and b is the bias value. The forget gate $\mathbf{f}_t$ has the capability of deciding what information will be "forgotten" and removed from the cell state $\mathbf{C}_t$. $\mathbf{i}_t + \mathbf{g}_t$ determines what type of information will be "added" to the cell state $\mathbf{C}_t$. The output $\mathbf{h}_t$ is determined by both $\mathbf{o}_t$ and the cell state $\mathbf{C}_t$.

3 PROPOSED CELLULAR IOT DEVICE IDENTIFICATION METHOD

3.1 Motivation

With the proliferation of IoT applications, the security of IoT devices has become an emergent issue as most of the IoT devices are less capable as non-IoT devices such as mobile phones and laptops. Most massive IoT devices have a Power Saving Mode (PSM) mechanism enabled to save energy such that devices could run a long time (months or years) without changing battery. However, these "special" features of IoT devices also exposes themselves to potential malicious attacks. A malicious attacker can take advantage of the different architectures of IoT devices and launch attacks such as data spamming attacks through successful identification of IoT devices [18].

Successful identification of the IoT devices is often the first step of launching more aggressive attacks. Therefore, an efficient and effective way of classifying cellular IoT devices from non-IoT devices are quintessential. In the TCP ACK packet-based probing method [18], a fundamental assumption is based on the fact that when there are probing packetse.g., the TCP ACK packets acknowledge the sequence number that has not been used yet by the other TCP connection end are sent out to IP addresses within a network that the receiver has to respond, all the active mode cellular IoT devices

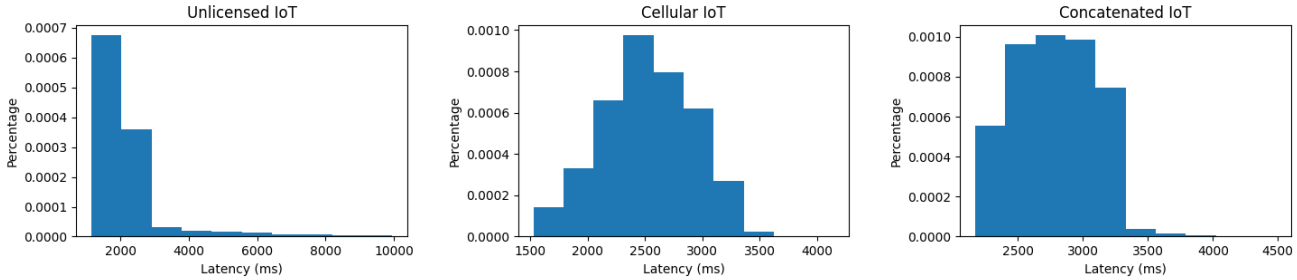

Figure 3: The end-to-end latency of IoT devices in three different network scenarios[11]

and online non-IoT devices will response to the probing TCP ACK packets. If no response packet is received after the probing packet is sent out, then the IP address will be inferred as a cellular IoT device that is sleeping in the PSM. In this assumption, the offline non-IoT device case is excluded because the probing packets will be sent out only when there exists observed traffic coming from that IP address.

However, based on the data analysis we conducted on real-world dataset, we found that things are more complicated in reality than we thought. There are two important factors that will complicate this situation:

End-to-end latency. There are various inferences in a cellular network that could lead to the latency of delivering the TCP ACK packets in a timely manner and even direct packet loss. When data is transmitted over a complicated network environment, including both wireless and wireline, there exists a latency and reliability problem. End-to-end latency, the time for a packet to be transmitted from the source host to the destination host, is often used to measure the latency. It generally consists of propagation delay, processing delay, and queuing delay. For massive IoT applications, devices usually have features like low-cost, low energy, and small data volumes, thus the delay can be enlarged if not well-managed, especially for the processing and queuing delay.

To show the latency problem, we use the dataset provided in [11] to demonstrate the end-to-end latency of 30,000 IoT data points. The latency distribution of the devices are shown in Fig. 3. Three different types of IoT scenarios are measured: unlicensed IoT, cellular IoT, and concatenated IoT. From the measured results, we can observe that for three different cases, the end-to-end latency can vary from around one thousand milliseconds to 10 thousand milliseconds. This kind of end-to-end latency will decrease the accuracy of the identification of IoT devices based on TCP ACK packets. The reason is because the attacker might receive an ACK response packet sent when the device was still in its active mode, but actually it is in sleeping mode when the attacker sends out new probing packets. In this situation, the device will not be classified as IoT devices. On the other hand, if this latency happens to non-IoT devices, it might mislead the attacker to classify non-IoT devices as cellular IoT devices when the attacker could not receive the response packets in time.

Alternation of operation modes. The other factor that complicates the classification process is the characteristics of operation modes of IoT devices. We analyzed a dataset that contains one month of binary activities of 4060 urban IoT nodes [5]. We found that the IoT nodes measured demonstrated a fixed pattern: it always differs from the status when measured in previous time slot, which means that the device is always alternating between active and inactive modes every time it is measured. The results are shown in Fig.4. This is reasonable for real-world scenarios as cellular IoT devices are massive IoT applications, which means that usually they are deployed massively over a wide area to collect data in a fixed pattern. Therefore, their change of operation modes can also be fixed.

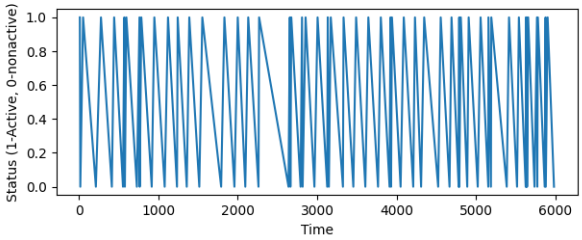

Figure 4: The alternation between active and inactive modes of IoT nodes [5].

However, the alternation of operation modes will complicate the probing process in [18]. When a probing attack is conducted in a fixed time interval series like 15s, 30s, 60s, 180s, 300s, if the IoT device is working in a fixed pattern of alternating modes between active and inactive mode, there is no guarantee that a probing packet will fall into a specific sleeping mode interval. We know that if we follow the probing intervals, we will have a packet falling into the sleeping mode, but we do not know which one will.

On the other hand, reliability is a problem that could be caused by packet contamination or loss during the transmission. This can also be enlarged given the low-cost feature of cellular IoT devices. Many of the IoT devices do not have the powerful capability of error correction like in non-IoT devices such as mobile phone and laptops. Combined with the network latency we identified above, this will make the probing attack much more difficult.

3.2 Attack architecture

Given the two factors we identified above that complicates the classification of the IoT devices from non-IoT devices in a cellular network, we propose a new machine learning based method to

identify the IoT devices more effectively. As we found from the analysis of real-world data, when an attacker sends out probing TCP ACK packets to the IP addresses within a cellular network, the response packets might not arrive at the attacker's side in a timely way, or they may be even get lost. However, as demonstrated in [13], IoT devices are typically less mobile than non-IoT devices, such as smartphones. This implies that the fundamental environment factors and patterns of the function of IoT devices are likely to remain consistent. If the environment changes, then the features are reflected sequentially. This kind of consistency and variation can be represented using the cell state in LSTM networks.

From the attacker's perspective, we propose an effective way of identifying cellular IoT devices from non-IoT devices using LSTM networks. After the IoT devices are detected, further attacks such as data spamming can be deployed [18]. The architecture of our proposed attack architecture is shown in Fig. 5. The main idea of the identification process is the attacker will keep sending TCP ACK probes to both IoT devices and non-IoT devices in a fixed pattern. It will collect the ACK packets it received from the network and use this data (including the failure information for those packets not responded) to train an LSTM-based neural network model. Then the learned model can be applied to classify new IP addresses when we interact with them in the same probing pattern we used while training. The advantage of using LSTM-based architecture is that we can use the cell state in the LSTM to represent and "memorize" the environment context and can be updated when it changes. Therefore, we have the flexibility of adapting to the dynamics of the probing context.

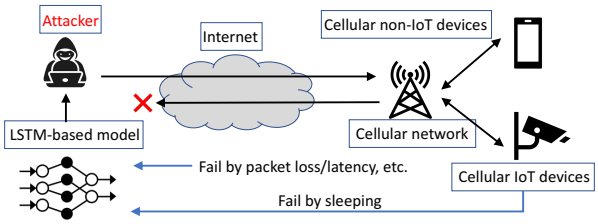

Figure 5: Our proposed attack architecture.

The problem can be specifically defined as follows. Given the data sequence $\mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, ..., \mathbf{x}_T\}$ that we collected while interacting with devices and the corresponding category information $y = \{y_1, y_2, y_3, ..., y_T\}$, we want to train a LSTM-based machine learning model $f : \mathbf{x} \rightarrow y$. During the training process, the attacker will send out TCP ACK packets in a fixed pattern. We want to ensure that the pattern is fixed for all probing packets during the training and identifying process. However, the pattern can be set by the attacker in any way that can fall into all the modes of the IoT devices. For example, a random sequence of length m can be generated within a time period and used as the time interval to send out probing packets.

Then the attacker collects the response packets it receives from the IP address and forms it into a feature vector as $\mathbf{x}_t = \{x_1, x_2, ..., x_m\}$. If the response of the IP address is not received, the value is denoted as 0. Thus, if no response is received for all the probing packets sent out during a time period, the ACK vector is given by $\mathbf{x}_t = \{0, 0, ..., 0\}$. If the IP address is a cellular IoT device, we denote the corresponding $y_t = 1$, otherwise, $y_t = 0$. Therefore, our goal is to train an LSTM model to learn the function $f : \mathbf{x} \rightarrow y$.

In the training process, we want to obtain enough samples from the cellular network to improve the accuracy of the learned LSTM model. The objective of the training process is to minimize the least square loss function: $\mathcal{L}(\mathbf{x}, \theta) = \min \sum_t \frac{1}{2}(y_t - \bar{y}_t)^2$, in which y_t is the true label while $\bar{y}_t$ is the output of the model. Other loss functions can also be used. In the test process, we adopt the same fixed pattern we use in the training process to sending out TCP ACK probing packets to IP addresses we want to evaluate or attack. After the response information of multiple probing packets are received, we can feed the data into the learned LSTM-based neural network model to perform classification.

4 EXPERIMENTAL VALIDATION

To validate the performance of our proposed attack method, we conduct the validation experiments based on two public datasets. The first dataset contains over 30,000 data points of IoT devices regarding the end-to-end latency due to the low-power wide-area network (LPWAN) inferences [11]. The second dataset contains one month of the binary activities of 4060 urban IoT nodes [5].

To utilize the two datasets for our experimental validation, we assume that when non-IoT IP addresses are probed, the attacker will receive the response packets with a random packet loss. The random packet loss rate is set as 1.31% [2]. For the cellular IoT devices, the status given in [5] consists of one month of activities for 4060 IoT nodes. We generate a fixed random interval pattern of dimension 9 to get 10 probing results, i.e., we make the decision about the types of IP addresses based on 10 probing results conducted. Every time when an IoT device is probed, we check out the binary status from the dataset [5]. If the status at that time is active, then we add a network end-to-end latency. The latency value is randomly sampled from the cellular IoT latency in [11]. If the latency is beyond 5 seconds [18], then we set the response as "0", which means the attacker assumes that the response is not received. Otherwise, it is set as response packet received, which is represented as "1".

From the the dataset in [5], we have the activity data of 4060 cellular IoT nodes over one month. We generate the response data as a 10000×11 matrix. For each data entry, the first 10 dimensions are used to represent the response status, and the 11th dimension is used to denote the type of the device. We use "1" to denote cellular IoT devices and "0" to denote non-IoT devices. To balance the training data, we generate the same number of data entries for the non-IoT devices. In the training process, we use the data of 1000 IoT devices and 1000 non-IoT devices to train the model and use the remaining data of 3060 IoT devices and 3060 non-IoT devices as test data. We conduct 10 epochs of training and implement the LSTM models in TensorFlow 2.4.0. The loss and accuracy results after each epoch for the training data and test data, respectively, are shown in Fig. 6. From the experimental results we observe that after 6 epochs of training, the model converges to the point that it can precisely classify all the test devices. It achieves the accuracy of 100%, which demonstrates that our methods could learn and identify the types of devices in an effective and efficient way.

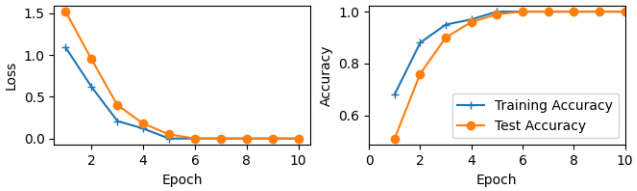

Figure 6: The loss and accuracy of training and test processes.

We also evaluate the performance when different number of training devices are used. Six groups of experiments are performed, in which data of 200,400,600,800,1000,1200 devices are used to train the model respectively. The remaining data is used as test data. Table 1 shows the number of epochs needed to achieve the 100% accuracy. It only takes 9 epochs to achieve 100% of training accuracy when only data of 200 devices are used, which validate from another perspective the efficiency of the LSTM-based identification method.

The size of training devices	Epochs required
200	9
400	9
600	7
800	6
1000	6
1200	4

Table 1: The number of epochs required to converge to 100% of training accuracy under different sizes of training devices.

5 POTENTIAL DEFENSES

IoT cellular devices can be identified by attackers mainly because they have higher requirements for the power consumption, that is why PSM is designed to save power of IoT cellular devices. However, it is this PSM that also exposes IoT cellular devices to various attacks. From the defender's perspective, to "hide" the IoT devices among normal non-IoT devices, one potential strategy is to remedy IoT cellular devices and reduce the time that IoT devices are running in PSM or make the PSM time unpredictable, thus making it difficult for attackers to identify the differences of IoT and non-IoT devices. A second strategy is to keep the design of IoT devices, but enable it to send back ACK packets when probed by request from online. The third strategy is to train a machine learning model on the non-IoT devices and make IoT cellular devices run in a similar style.

6 CONCLUSION

Cellular IoT technologies have been deployed widely with the proliferation of IoT devices. Massive IoT services supported by cellular networks are emerging to play an important role in IoT applications. However, the characteristics of low-cost, low energy and small data volumes in cellular IoT devices make them less capable of defending themselves against attacks. In this paper, we developed an effective and efficient attack method based on a special type of TCP ACK probing packets. The response information, especially those failed, is collected and embedded into a feature vector. We trained an LSTM-based neural network model to identify cellular IoT devices from the cellular network. We evaluated this method on public datasets and achieved identification accuracy of 100% within less than 10 epochs of training, which validates the efficiency and effectiveness of the proposed method. The results raise the need for robust defense for the IoT devices against detection by the attackers.

ACKNOWLEDGEMENTS

The research at USF was supported in part by NSF CNS-2044516.

REFERENCES

[1] GSM Association et al. 2018. LTE-M Deployment Guide to Basic Feature Set Requirements.
[2] Peter Benko and Andras Veres. 2002. A passive method for estimating end-to-end TCP packet loss. In *IEEE Global Telecommunications Conference*, Vol. 3. 2609–2613.
[3] Kemal Davaslioglu, Sohraab Soltani, Tugba Erpek, and Yalin E Sagduyu. 2019. DeepWiFi: Cognitive WiFi with deep learning. *IEEE Transactions on Mobile Computing* 20, 2 (2019), 429–444.
[4] Tugba Erpek, Timothy J O'Shea, Yalin E Sagduyu, Yi Shi, and T Charles Clancy. 2020. Deep learning for wireless communications. *Development and Analysis of Deep Learning Architectures* (2020), 223–266.
[5] Arvin Hekmati, Nishant Jethwa, Eugenio Grippo, and Bhaskar Krishnamachari. 2023. Large-Scale Urban IoT Dataset. https://doi.org/10.21227/dtfr-b327
[6] Tao Hou, Tao Wang, Zhuo Lu, Yao Liu, and Yalin Sagduyu. 2021. IoTGAN: GAN powered camouflage against machine learning based IoT device identification. In *IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*. IEEE, 280–287.
[7] M Laner, N Nikaein, P Svoboda, M Popovic, D Drajic, and S Krco. 2015. Traffic models for machine-to-machine (M2M) communications: types and applications. In *Machine-to-machine (M2M) Communications*. Elsevier, 133–154.
[8] Manuel Lopez-Martin, Belen Carro, Antonio Sanchez-Esguevillas, and Jaime Lloret. 2017. Network traffic classifier with convolutional and recurrent neural networks for Internet of Things. *IEEE access* 5 (2017), 18042–18050.
[9] Zhengping Luo, Shangqing Zhao, Zhuo Lu, Yalin E Sagduyu, and Jie Xu. 2020. Adversarial machine learning based partial-model attack in IoT. In *ACM Workshop on Wireless Security and Machine Learning*. 13–18.
[10] Dinh C Nguyen, Ming Ding, Pubudu N Pathirana, Aruna Seneviratne, Jun Li, Dusit Niyato, Octavia Dobre, and H Vincent Poor. 2021. 6G Internet of Things: A comprehensive survey. *IEEE Internet of Things Journal* 9, 1 (2021), 359–383.
[11] Alvin Ramoutar, Zohreh Motamedi, and Mouhamed Abdulla. 2021. Latency of Concatenating Unlicensed LPWAN with Cellular IoT: An Experimental QoE Study. In *IEEE Vehicular Technology Conference (VTC2021-Fall)*. IEEE, 1–6.
[12] Yalin E Sagduyu, Yi Shi, and Tugba Erpek. 2019. IoT network security from the perspective of adversarial deep learning. In *IEEE International Conference on Sensing, Communication, and Networking (SECON)*. 1–9.
[13] M Zubair Shafiq, Lusheng Ji, Alex X Liu, Jeffrey Pang, and Jia Wang. 2013. Large-scale measurement and characterization of cellular machine-to-machine traffic. *IEEE/ACM transactions on Networking* 21, 6 (2013), 1960–1973.
[14] Alex Sherstinsky. 2020. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Physica D: Nonlinear Phenomena* 404 (2020), 132306.
[15] Arunan Sivanathan, Hassan Habibi Gharakheili, Franco Loi, Adam Radford, Chamith Wijenayake, Arun Vishwanath, and Vijay Sivaraman. 2018. Classifying IoT devices in smart environments using network traffic characteristics. *IEEE Transactions on Mobile Computing* 18, 8 (2018), 1745–1759.
[16] Ralf C Staudemeyer and Eric Rothstein Morris. 2019. Understanding LSTM–a tutorial into long short-term memory recurrent neural networks. *arXiv preprint arXiv:1909.09586* (2019).
[17] Mojtaba Vaezi, Amin Azari, Saeed R Khosravirad, Mahyar Shirvanimoghaddam, M Mahdi Azari, Danai Chasaki, and Petar Popovski. 2022. Cellular, wide-area, and non-terrestrial IoT: A survey on 5G advances and the road toward 6G. *IEEE Communications Surveys & Tutorials* 24, 2 (2022), 1117–1174.
[18] Sihan Wang, Guan-Hua Tu, Xinyu Lei, Tian Xie, Chi-Yu Li, Po-Yi Chou, Fucheng Hsieh, Yiwen Hu, Li Xiao, and Chunyi Peng. 2021. Insecurity of operational cellular IoT service: new vulnerabilities, attacks, and countermeasures. In *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*. 437–450.
[19] Shangqing Zhao, Zhengping Luo, Zhuo Lu, Xiang Lu, and Yao Liu. 2017. Stateful Inter-Packet Signal Processing for Wireless Networking. In *Annual International Conference on Mobile Computing and Networking*. 15–27.