Journal of Cybersecurity and Privacy

MDPI

Article

RSSI-Based MAC-Layer Spoofing Detection: Deep Learning Approach †

Pooria Madani * and Natalija Vlajic

Electrical Engineering and Computer Science, York University, Toronto, ON M3J 1P3, Canada; vlajic@eecs.yorku.ca

* Correspondence: madani@eecs.yorku.ca

† This paper is an extension version of the conference paper: Madani, P.; Vlajic, N.; Sadeghpour, S. MAC-Layer Spoofing Detection and Prevention in IoT Systems: Randomized Moving Target Approach. In Proceedings of the 2020 Joint Workshop on CPS & IoT Security and Privacy, Virtual Event, USA, 9 November 2020.

Abstract: In some wireless networks Received Signal Strength Indicator (RSSI) based device profiling may be the only viable approach to combating MAC-layer spoofing attacks, while in others it can be used as a valuable complement to the existing defenses. Unfortunately, the previous research works on the use of RSSI-based profiling as a means of detecting MAC-layer spoofing attacks are largely theoretical and thus fall short of providing insights and result that could be applied in the real world. Our work aims to fill this gap and examine the use of RSSI-based device profiling in dynamic real-world environments/networks with moving objects. The main contributions of our work and this paper are two-fold. First, we demonstrate that in dynamic real-world networks with moving objects, RSSI readings corresponding to one fixed transmitting node are neither stationary nor i.i.d., as generally has been assumed in the previous literature. This implies that in such networks, building an RSSI-based profile of a wireless device using a single statistical/ML model is likely to yield inaccurate results and, consequently, suboptimal detection performance against adversaries. Second, we propose a novel approach to MAC-layer spoofing detection based on RSSI profiling using multi-model Long Short-Term Memory (LSTM) autoencoder—a form of deep recurrent neural network. Through real-world experimentation we prove the performance superiority of this approach over some other solutions previously proposed in the literature. Furthermore, we demonstrate that a real-world defense system using our approach has a built-in ability to self-adjust (i.e., to deal with unpredictable changes in the environment) in an automated and adaptive manner.

Keywords: IoT security; spoofing; MAC authentication; intrusion detection system; LSTM autoencoders

Citation: Madani, P.; Vlajic, N. RSSI-Based MAC-Layer Spoofing Detection: Deep Learning Approach. *J. Cybersecur. Priv.* **2021**, *1*, 453–469. https://doi.org/10.3390/jcp1030023

Academic Editors: Phil Legg and Giorgio Giacinto

Received: 20 May 2021
Accepted: 29 July 2021
Published: 12 August 2021

1. Introduction

The proliferation of the Internet of Things (IoT) and Wireless Sensor Network (WSN) networks has revived an old yet serious form of attack—*MAC-layer Spoofing* or also referred to as *Identity Spoofing*. In MAC address spoofing attack, as the name suggests, a rouge wireless node masquerades as another legitimate device by cloning the legitimate device's MAC address. Identity spoofing, in general, is a precursor for packet injection (another well-known type of attack) and thus requires careful consideration as part of any sound defense plan.

The most common way of defending against this form of attack is through the use of cryptographic techniques for MAC-address authentication [1]. Unfortunately, due to the resource limitations that are inherently present in many IoT and WSN devices (e.g., low processing power, low memory capacity, and limited battery life), many of these devices operate with very scaled-down (if any) versions of encryption and authentication protocols. For example, it is discovered that due to ease-of-installation by non-technical consumers, Philips IoT Smart Bulbs do not employ any form of encryption and authentication as

specified by 802.15.5 protocol standard [2]. Or, in the case of a multihop WSN, the intermediate relaying nodes generally do not engage in the verification of the authenticity of the *relayed data frames*—authenticity verification of these frames takes place only at the final (i.e., destination) node. Authentication by intermediate nodes is typically omitted in order to reduce the nodes' energy consumption as well as minimize the possibility of a *battery exhaustion* attack [3] (readers should review seminal work by Nguyen et al. [4] for a complete survey of energy depletion attacks against low power wireless networks).

It should be noted that in a number of standardized wireless protocols that are still in use today, cryptographic authentication is simply not intended for all stages/frames of a communication process. For example, in all variants of IEEE 802.11 preceding 802.11 w, only data frames are protected, while control and management frames are used without any protection [5]. Thus, one should make provisions for extra security measures when cryptographic authentication is not supported by protocols deployed within certain application domains.

Clearly, in wireless systems with limited cryptographic and authentication protection, other alternative measures against MAC address spoofing are required. One such measure—which can also be used as an added layer of security even in wireless systems with extensive cryptographic and authentication protection—is the utilization of physical layer (i.e., signal-level) parameters. *Received Signal Strength Indicator* (RSSI) is a wireless communication variable that is directly influenced by the transmission power and the location of the transmitter as well as different environmental variables such as obstacles. As suggested in a number of earlier research works (e.g., [6–8]), RSSI values can be used to create the *fingerprint profile* of each device in a wireless network and then deploy these profiles to do a preliminary authenticity check against MAC spoofing attacks. Another point that makes RSSI profiling an attractive ally against MAC spoofing attacks is that the use of this single real-valued physical-layer variable is easy to implement, requires no modifications to existing higher layer protocols and applications, and has a very small processing and memory footprint.

There have been many research works in the past investigating the use of RSSI profiling for the purpose of MAC spoofing detection (some of which are surveyed in Section 2). Most of these works implicitly assume that: (1) RSSI samples received from a non-moving transmitting device form a stationary time-series with normally distributed variance, and (2) RSSI values are *independent and identically distributed* (i.i.d.) samples from an unknown normal distribution. Moreover, in the given works, the act of profiling a wireless device based on its RSSI values strongly relies on these two very assumptions. However, in our recently conducted study, the two assumptions (RSSI samples are stationary and i.i.d.) have come under scrutiny. Namely, through our extensive real-world experimentation, we have observed that RSSI values measured by a receiving node are highly affected by changes (e.g., moving objects) in their operating environments. In particular, we have observed that moving human bodies (and their absence) have a noticeable effect on RSSI values of IoT devices deployed in a residential environment, and as a result the variance of the RSSI time-series changes significantly when occupants are present and move around the property—we call this effect *time-series clustering* [9] (refer to Figure 1 where there are two different clusters, one with lower volatility than the other). Furthermore, it is clear from the depicted figure that there is a correlation between neighboring RSSI values; therefore, it would be hard to justify the claim that neighboring RSSI values are independent (as presumed by previous works [6–8]).

Except in a few usage cases where there are no moving objects in the environment (e.g., farmland monitoring), most real-world IoT networks deploy computing/sensing nodes in environments with some number of movable objects. Thus, in order to account for changes in RSSI values due to the above described clustering effect, it is necessary to have an adaptive and/or multi-model RSSI-based profiling scheme that will be able to improve/reduce the rates of false positives (in our previous work [10], we demonstrated how i.i.d. assumption pertaining to RSSI values can lead to probable evasion of detection

systems that rely on RSSI-based profiling). In this work, we have proposed and studied a multi-classifier system to profile IoT devices based on their RSSI values under two moving object conditions (presence vs. absence of objects in the surrounding environment). Also, our profiling approach takes into consideration the relationship between neighboring RSSI values in the time-series to further improve the accuracy and robustness of IoT node profiles.

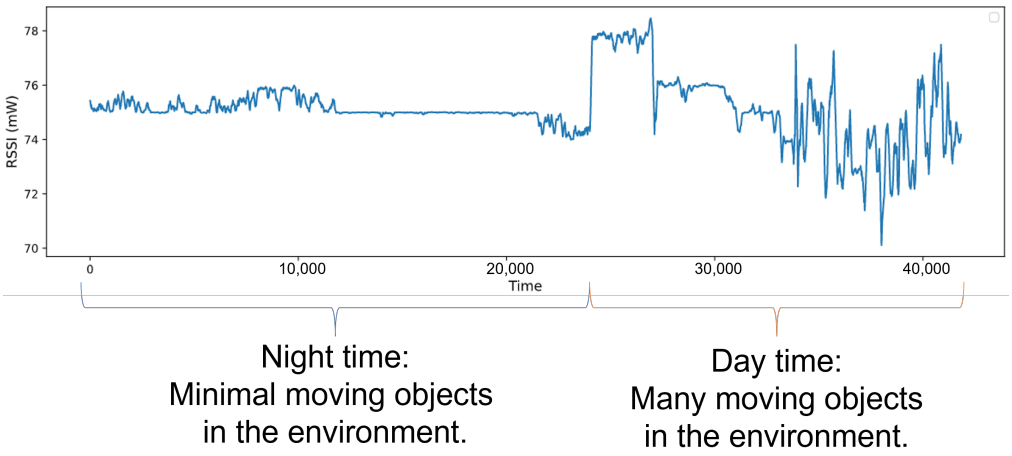

Figure 1. RSSI values of an IoT device deployed in a residential property with routine movements of occupants in a 24 h period.

The content of this paper is organized as follows: In Section 2, we discuss some of the notable previous works in RSSI-based MAC address spoofing detection. In Section 3, we present the threat model and the main assumptions about the adversary's capabilities as pertaining to our work. In Section 4, we propose our LSTM-based (Long Short-Term Memory) profiling scheme that has been devised to detect and classify MAC-spoofing traffics. In Section 5, we discuss adversarial traffic generation used to test the robustness of our approach and compare the effectiveness of our approach with the state-of-the-art RSSI based approaches previously proposed to deal with adversarial attacks.

2. Related Works

Wireless MAC Address Spoofing Detection is a well-studied topic in the literature on Wi-Fi and Wireless Sensor Networks. In the seminal paper [11], Faria and Cheriton were among the first ones to propose the use of RSSI values as a fingerprinting variable to detect MAC spoofing attacks in a WLAN environment. As part of their detection model, it is assumed that there are multiple access points (APs) capable of receiving the wireless signals from all clients in the network, so the RSSI values measured at each AP's antenna and for each transmitter are ultimately aggregated into a single profile. Consequently, a masquerading attack is detected by comparing the aggregated RSSI values of two consecutive data frames with the same MAC identifier. Also, they have demonstrated that using multi-sensing APs, and assuming constant transmitting power, a physical node can be triangulated with an accuracy of 5 to 10 m. Unfortunately, the practical merit of these findings is rather limited since the use of multiple overlapping APs in many WSN and IoT networks is not always possible.

Chen et al. [12] and Wu et al. [6] have both independently proposed the use of *k-means* clustering algorithm to detect signal/frame spoofing by a rogue access point (AP). Their work is grounded on the assumption that the sequence of last n RSSI values received from an AP would have minimum fluctuations around the mean in the absence of another rogue AP (i.e., an 'Evil Twin'). Thus, when clustering the elements of a received RSSI sequence into two clusters using k-means algorithm in the absence of an Evil Twin, the distance between two formed centroids would be small (i.e., smaller than a threshold value). At the

same time, a large distance between the centroids of the two formed clusters would be indicative of the existence of an Evil Twin AP with its unique RSSI distribution. However, since this approach does not involve any offline learning (i.e., a previously trained model of what should be considered a legitimate distribution), the MAC address spoofer and the legitimate node must transmit in relatively close time intervals for the detection to actually work.

Sheng et al. [13] studied the effect of antenna diversity in 802.11 access points and their effect on RSSI device fingerprinting as well as spoofing detection. They demonstrated that RSSI values from a stationary receiver collected at a stationary transmitter form a mixture of two Gaussian distributions due to antenna diversity permitted under 802.11 protocol. As a result, they have trained a Gaussian mixture model for each wireless node and access point pair in the network and used a log-likelihood ratio test on the sequence of latest received RSSI at each access point from a given MAC address. A transmitting node is ruled spoofed if the ratio test fails by more than n Gaussian mixture models—where n is smaller than the number of available access points in the network and needs to be set empirically. However, using available off-the-shelf hacking tools an adversary can easily manipulate its transmission power to evade detection by this model, as discussed in later sections.

Gonzales et al. [14] have developed a novel technique known as context-leashing for the detection of public Evil Twin access points. They have argued that publicly available access points such as the ones available at franchise coffee shops (e.g., Starbucks) share service set identifiers (SSID) across different locations and oftentimes lack any authentication. This provides an opportunity for adversaries to spoof such SSIDs and trick clients into associating with a rogue access point (e.g., after performing a dissociation attack). The defense against the Evil Twin APs proposed in [8] assumes the use of a so-called context-leashing engine. Upon association with a publicly available access point, the context-leashing engine would collect a list of context $C_i = \{(c_1, r_1), \ldots, (c_n, r_n)\}$, which contains the list of all visible SSIDs (denoted by c_j, $j = 1, \ldots, n$) and their corresponding average RSSI values (denoted by r_j, $j = 1, \ldots, n$) that is reachable at the time of association with a particular SSID in the environment. For any future reassociation with a given SSID, a new context list is constructed and compared to the previously stored one. If the context-list of available neighboring SSIDs and their average RSSI values does not have a significant (empirically defined) overlap with the historical context-list, then the associated SSID is deemed an Evil Twin and the connection should be terminated. The main drawback of their method is the assumption that the list of SSIDs in a given geolocation remains relatively unchanged over time. However, with today's tethering capabilities of cellphones, this assumption is far from the truth.

3. Threat Model and Assumptions

In this section, we introduce the main annotation and assumptions of our work, which are also illustrated in Figure 2. First, consider a simple setup where there are a legitimate transmitting node (e.g., a temperature sensor) denoted by s and a legitimate receiving node (e.g., an IoT hub) denoted by r communicating over a wireless channel. Also, we assume that r utilizes an arbitrary approach (including what we propose in this work) to profile s based on RSSI samples, it has received in a period absent of any adversary, and then uses this profile at runtime to differentiate between received *data frames* that carry s' MAC address (legitimate vs. spoofed ones). Finally, let α denote the adversary with the following characteristics:

- The adversary is situated at a location from which it can observe/receive signals transmitted by all legitimate senders (when sending data frames) and receivers (when sending acknowledgment frames back) in the given network.
- The adversary is aware of the transmission power setting (P_{Tx}) of the legitimate sender(s), which is not a substantial assumption as system information about most IoT/WSN devices is publicly accessible on the Internet.

- The adversary has no prior knowledge of the actual physical/geographic locations of other (legitimate) nodes in the network.
- Network participants, including the adversary, are equipped with regular/common omnidirectional antennas, and are not capable of detecting the positional angle of the transmitting nodes. However, the adversary can move about in order to triangulate other nodes' locations based on the strength of the signal received from those nodes [10].
- The adversary itself is an *active* node capable of adjusting its transmission power.
- The adversary is also capable of altering (i.e., spoofing) its MAC address value—i.e., it can generate data frames that carry MAC addresses of other legitimate nodes from this particular network.

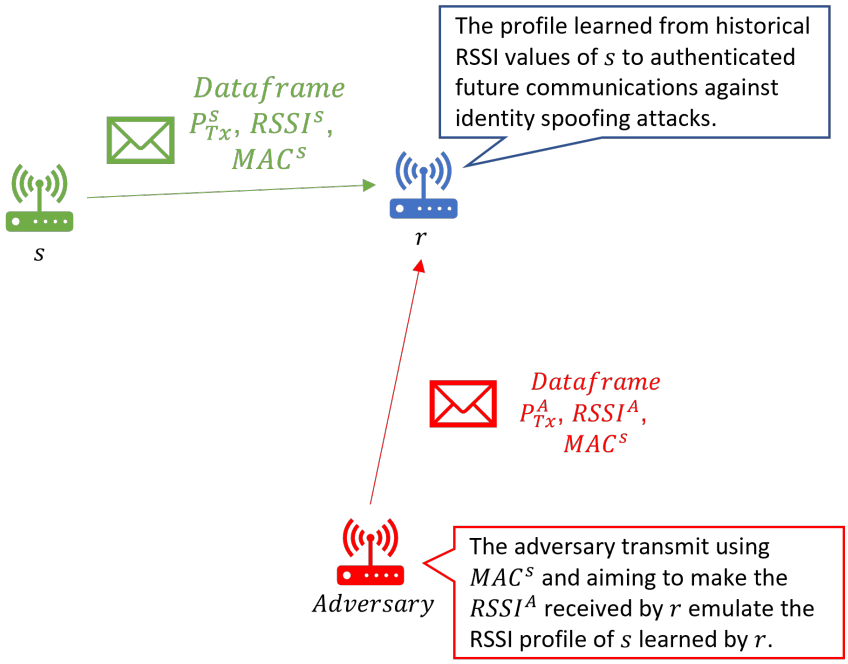

Figure 2. Overview of the threat model: the goal of the receiving node is to use historical (clean) RSSI values from the legitimate sender to learn a robust profile to use in future against identity attacks; while the goal of the adversary is to get past the established profile by taking over s identity.

The ultimate goal of adversary α is to impersonate a particular s by transmitting frames with s' spoofed MAC address. The spoofed frames are specifically intended for a particular r. Since, according to the assumptions of our work, the transmitter's RSSI values are registered and used by r for the purposes of MAC-spoofing detection, the adversary first needs to discover/adjust its transmission power (P_{Tx}) such that its spoofed frames (when received by r) get accepted as genuine with a high probability—i.e., some desired probability of evasion is achieved by the adversary. This particular problem—of how to discover/adjust the transmission power so as to achieve a certain evasion probability—is closely related to the optimal adversarial evasion problem introduced by Nelson et al. [15] and further extended by Madani and Vlajic [10] to the IoT realm.

4. Detection Approach: Deep Authentication

As demonstrated in Figure 1 (and argued in Section 1), given that RSSI time-series values of a wireless IoT device are not i.i.d., one could incorporate dependencies among neighboring RSSI values to build more robust and accurate predictive models for the purpose of device authentication. Deep autoencoders are deep generative neural networks that have demonstrated a strong capability of modeling latent variables in anomaly detection and authentication datasets [16]. LSTM autoencoders [17], in particular, are known for their

generative modeling capabilities on time-series data. In this section we present our novel technique for authentication of legitimate IoT nodes using RSSI-based anomaly detectors deploying LSTM autoencoders. In addition, expanding on our argument from Section 1 with respect to the *time-series clustering-effect* of RSSI values in dynamic environments, we also discuss how our novel multi LSTM autoencoder architecture is able to switch between multiple trained LSTM models at runtime. Such a multi-LSTM autoencoder architecture would help with addressing the clustering effect of RSSI time-series.

4.1. LSTM Autoencoder Anomaly Detector

In the context of our work, let $X =< x_1, x_2, \ldots, x_n >$ denote an ordered sequence of n RSSI values received by node s. Then, the LSTM autoencoder is trained to learn two functions, namely, encoder $E(.)$ and decoder $D(.)$ such that $X \approx D(E(X))$. In other words, as depicted in Figure 3, the LSTM autoencoder learns an encoding state that best describes the structure of the training/input data and a decoding function that reconstruct the input sequence given the encoding state with minimal error. In general, large reconstruction errors occur when the input does not conform to the structure previously learned by the LSTM autoencoder. As such, a large reconstruction error can be used as a measure of input anomaly [16,18–20].

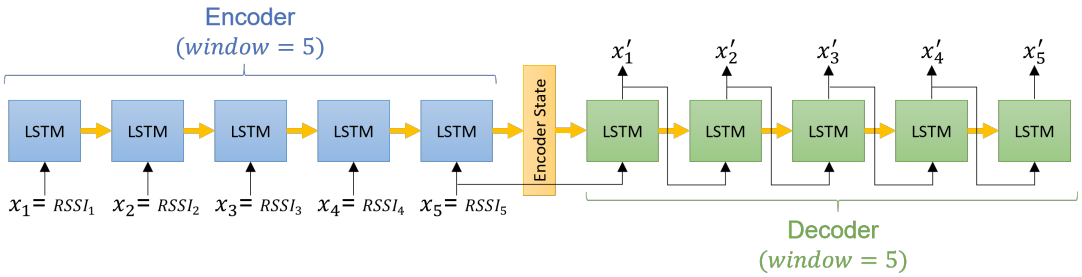

Figure 3. Anatomy of the LSTM autoencoder.

In order to build an RSSI profile of s (through the use of LSTM autoencoder), the receiving node r begins the process of collecting and assembling a time-series of RSSI values extracted from the data frames transmitted by s. Then, using a *rolling* window of size n, the time-series is segmented into m different overlapping sequences (where the extent of the overlap is controlled by the shift constant of the rolling window), which are further used to train the LSTM autoencoder. Since the LSTM autoencoder is supposed to learn the reconstruction of the input sequences, the m training inputs are also supplied as the expected outputs to the training algorithm with the *mean squared error* (MSE) as the loss function.

At runtime (i.e., during the actual use of the trained LSTM autoencoder for the purpose of attack/anomaly detection), n most recently observed RSSI samples are supplied into the trained LSTM autoencoder and then the MSE of the reconstructed sequence (relative to the provided input) is computed. Our experimental investigations (as described in Section 5) have demonstrated that the MSEs of the training data, in the absence of attack/spoofed instances, form a normal distribution. Therefore, our system uses Z-score to measure deviation from the expected MSE as the decision function to differentiate between the spoofed and the normal traffic. Specifically, for a Z-score $\geq l$ the system declares the inspected RSSI window as malicious, where l can be computed experimentally and set for the desired *false positive rate*.

4.2. Multiclassifer and Model Switching

As discussed in Section 1, in IoT environments with moving objects (e.g., residential or commercial premises), the RSSI time-series of a transmitting node can be divided into two significantly different time-series with substantially different volatility (i.e., time-series with clustering effect). Using the entirety of such a time-series (refer to Figure 1) for

the training of our system's LSTM autoencoder would result in a less sensitive anomaly detection model. Thus, we propose to deploy/train two independent LSTM autoencoders—one for the volatile period of the observed time-series when moving objects are present, and one for the relatively calm period when the relative volatility is at its minimum.

Now, one obvious issue that would have to be adequately addressed in an anomaly detection system with two LSTM autoencoders is the issue of their scheduling. As one possible approach, the system operator could manually set the exact time when each of the trained LSTM autoencoders is to be deployed according to his/her knowledge of the environment. However, in such a system with manually determined 'switch times', a number of potential problems could arise. For example, an employee of a factory showing up earlier than usual could significantly affect the RSSI time-series of the nearby sensors/transmitters, which as a result could trigger a false positive alert (provided the detection model corresponding to the non-volatile conditions is still active).

One way to resolve the above challenges is by simultaneously monitoring MSE Z-scores output by the two models at runtime, and looking for the point in time when the Z-score of one of the models crosses another. For example, as shown in our experimentation and depicted in Figure 4, at night where there are fewer moving objects in the environment, the night's LSTM autoencoder model is reconstructing the RSSI time-series perfectly as reflected by its low Z-score, while at the same time the day's LSTM autoencoder does a poor job in reconstructing the same RSSI time-series. However, during the transition period when moving objects start to appear in the environment, the night's LSTM autoencoder performance starts to decline, while the performance of the day's LSTM autoencoder (which is trained to cope with daytime volatility) starts to exhibit noticeable improvement with respect to the reconstruction MSE. Thus, the moment when the two Z-score time-series cross over each other would be the optimal point in time when the system should switch from using the nighttime to using the day-time LSTM autoencoder model. This suggests that by simply monitoring the output of both trained LSTM autoencoder models, it is possible to determine the optimal 'switch time' in an adaptable and automated manner.

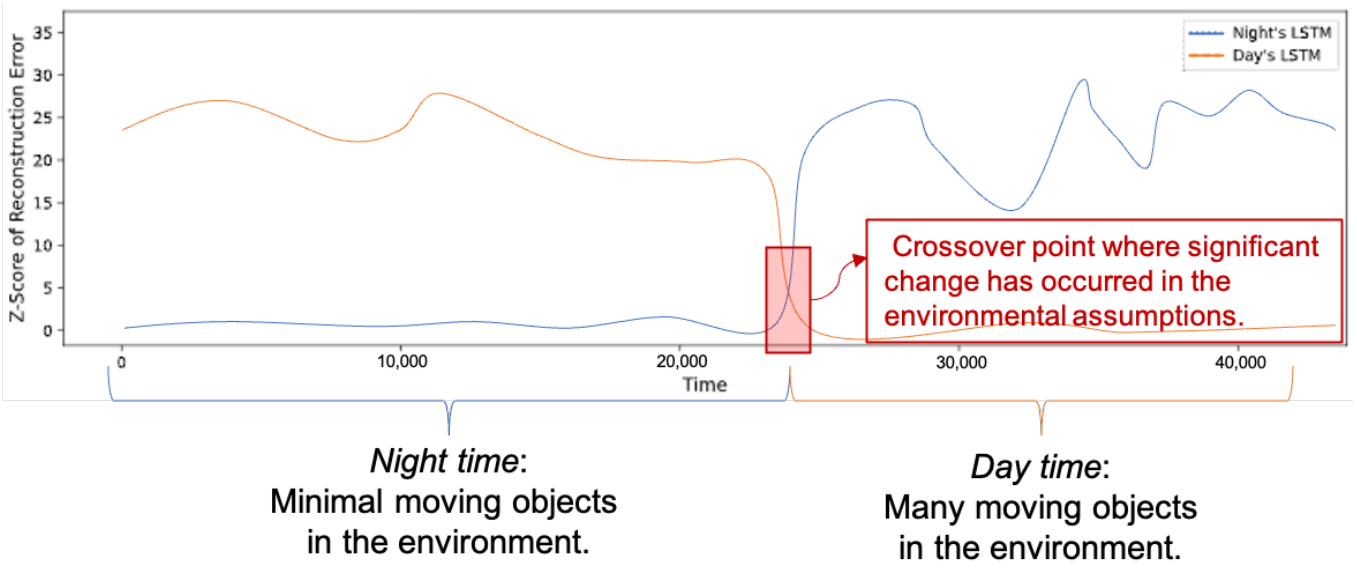

Figure 4. Starting at midnight, the Z-scores of reconstructed RSSI values corresponding to the transmitting node s using the two trained models (for day and night) are tracked. At about dawn, when occupants started to wake up and move about, the error rate of the night model significantly increases while the day model's error rate drops significantly.

5. Experiments and Results

5.1. Environment Setup

We have designed two experiments involving different forms of obstacles and moving objects to best collect the noise and other disturbances that IoT devices may face when attempting to profile their neighboring nodes using RSSI observations. In our experiments we have used three Digi XBee 3 Series programmable modules implementing IEEE 802.15.4. [21] (as depicted in Figures 5 and 6), where one device acts as the legitimate temperature reading sensor (denoted by s) transmitting its reading to the legitimate receiver (denoted by r) and the adversary (denoted by a) who spoofs the s' MAC address in the hope of providing false temperature readings to r.

In the first experiment (refer to Figures 6 and 7), s is situated in a waterproof container on the lawn outside the house, while r is situated in the second-floor bedroom. Aside from 5 occupants living on the property that move about the house during the day, outside pedestrians and moving vehicles affect s' RSSI values observed by r. The adversary is free to move about, both inside the property and outside, to carry out its spoofing attack (this is a very generous assumption to highlight a worst case scenario and superiority of our approach. In most settings, there is some degree of physical security that constrains adversaries in their physical positioning). This is an ideal experiment for resembling scenarios where IoT devices are separated by exterior walls and experience some degree of moving objects during the course of their daily operations.

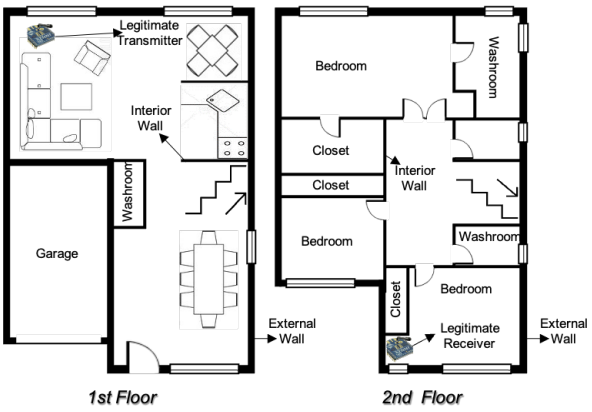

Figure 5. The legitimate transmitter is situated in the first floor family room while the legitimate receiver is situated in the second floor's bedroom separated by interior walls and an interior floor. The 5 occupants in the property are considered to be the influencing moving objects.

Figure 6. The legitimate transmitter is situated outdoors on the lawn transmitting temperature readings and the receiver is situated in the bedroom of the second floor separated by exterior building walls. The pedestrians and motor vehicles in the nearby residential area as well as the 5 occupants in the property are considered to be the influencing moving objects.

In the second experiment both r and s are situated in the property separated by a floor/ceiling and interior walls (depicted in Figure 6) while the adversary is allowed to move about inside and outside of the property. Similar to the first experiment the house occupants have their routine daily schedule of moving around the property during the day and resting (i.e., minimal movement) at night.

In the second experiment both r and s are situated in the property separated by a floor/ceiling and interior walls (depicted in Figure 5) while the adversary is allowed to move about inside and outside of the property. Similar to the first experiment, the house occupants have their routine daily schedule of moving around the property during the day and resting (i.e., minimal movement) at night.

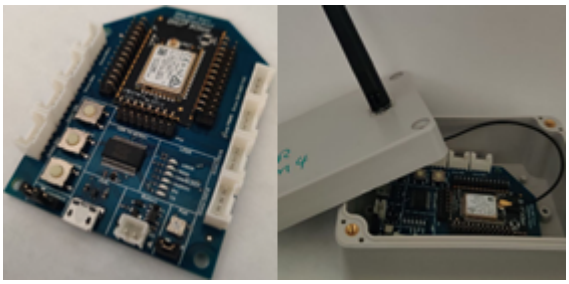

Figure 7. Digi XBee 3 Series programmable module implementing IEEE 802.15.4. in a weatherproof secure enclosure protecting the devices from the elements when deployed.

In both experimental setups, r starts its training phase by collecting RSSI samples from s (refer to Figures 8 and 9) both during hours of minimal and significant movements (24 h of capture of RSSI at the sample rate of 1 frame/s)—where these hours are assumed to be empty of any adversarial presence to perturb the training dataset. Once the training stage is completed, r starts using its two trained LSTM autoencoder models to authenticate received signals and detect MAC-spoofed frames (each LSTM autoencoder has 2 LSTM layers with 20 nodes each and a final dense lake of size 1 and using Adam [22] optimizer for training).

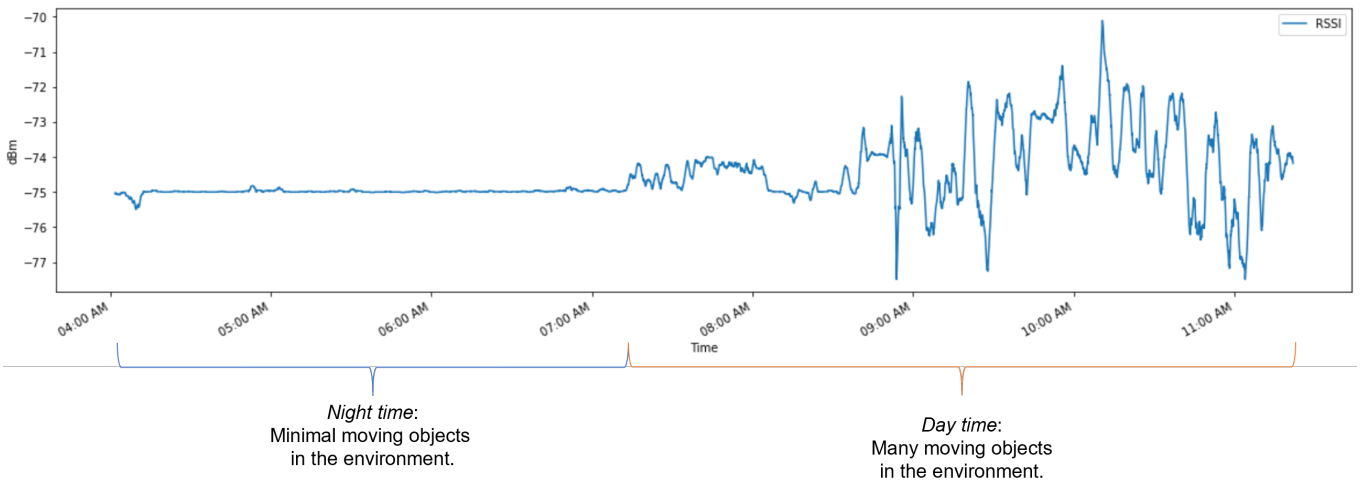

Figure 8. s' RSSI stream received by r during s' deployment outside of the property.

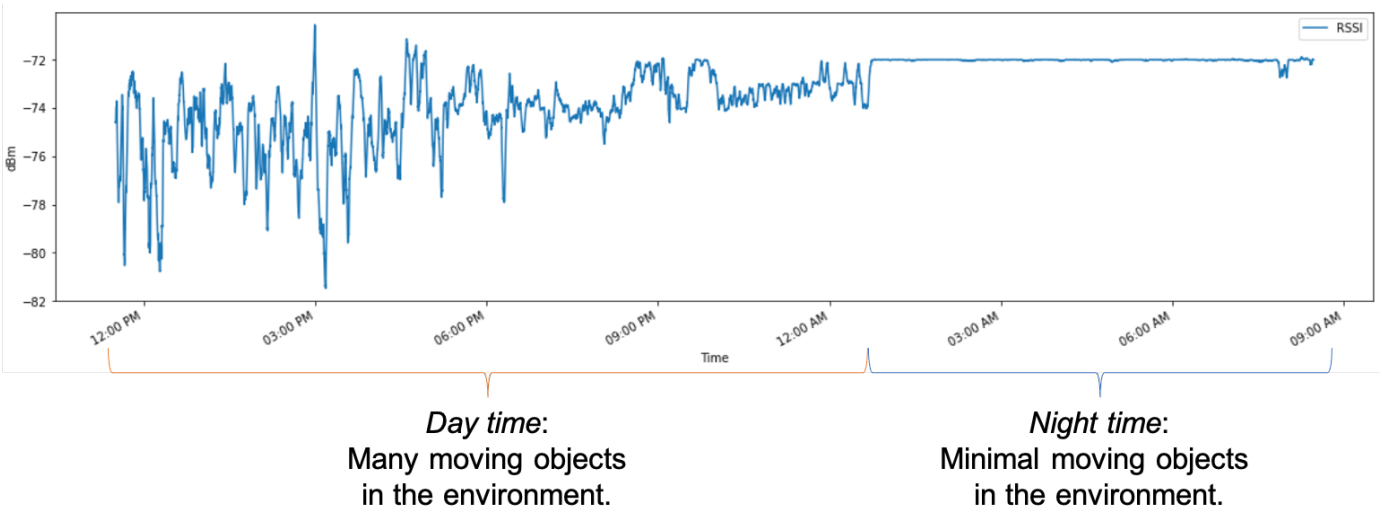

Figure 9. s' RSSI stream as received by r during s' deployment inside of the property.

5.2. Note on Special Spoofed Traffic Mix

All the surveyed works in Section 2 that use a rolling window on collected RSSI stream(s) for the purposes of signal classification (i.e., authentication) have implicitly assumed that each window of length n may fully consist of RSSI values from either an attacker or a legitimate device. However, this is not a realistic assumption given the unknown motivation and capabilities of adversaries. Moreover, many modern-day IoT devices (e.g., especially those used in home automation) are not battery operated and/or are not much concerned with energy preservation and as a result may be in frequent communication with other nearby devices. Consequently, in any given window of length n (used by the classification engine) there may exist some mix of the legitimate node's and the adversary's RSSI values as depicted in Figure 10.

(**a**) (**b**)

Figure 10. (**a**) Case where the adversary starts transmitting right after the legitimate node terminated its transmission; (**b**) The adversary gains access to the channel while the legitimate node has not finished transmitting all of its frames.

5.3. Model Classification Performance

We have evaluated our novel spoofing detection approach against the Support Vector Machine (SVM) one-class anomaly detection technique described in [23] (as a baseline detection model) and the state-of-the-art Log-likelihood ratio test approach proposed in [13]. We have evaluated all three approaches against two real-world datasets (refer to Section 5.1) using 10-fold cross validation. The average classification/detection performance is reported in Table 2.

We have trained two classifiers for our autoencoder as well as each of the other two approaches (SVM [23] and Log-likelihood [13]): one for the period of high volatility (e.g., environmental moving objects—daytime) and another for the period of low volatility (e.g., minimal environmental moving objects—nighttime) as reported in Table 1. All three classifiers perform relatively better during the low volatility period (i.e., nighttime) than the high volatility period—with our approach performing the best in both categories significantly.

Table 1. Summary of Related Works.

	Methodology	Shortcomings
Faria and Cheriton [11]	Using multiple access point recording RSSI values of individual nodes in the network and compare them with historical records and vote on authenticity of the given transmission.	The assumption of the existence of multiple APs is not realistic in many IoT and WSN applications. Using their approach a single AP can be easily evaded as discussed in Madani and Valjic [10]. Also, they did not entertain the existence of variable noises as a result of moving objects in the environment during different time periods.
Chen et al. [12]	Using k-means clustering and comparing cluster centroids distance to find existence of anomalies in RSSI values.	Treating a sequence of RSSI as identically distributed and independent observations. In Sections 1 and 5.2 we have discussed in detail why such assumptions are wrong and can be advantageous to the adversary.
Wu et al. [6]	Using k-means clustering and comparing cluster centroids distance to find existence of anomalies in RSSI values.	Treating a sequence of RSSI as identically distributed and independent observations. In Sections 1 and 5.2 we have discussed in detail why such assumptions are wrong and can be advantageous to the adversary.
Sheng et al. [13]	Uses Gaussian mixture models to model observed RSSI from a given node and create a normal/expected RSSI profile.	Capturing diversity caused by antenna diversity implemented by wireless nodes. Although did not entertain the existence of variable noises as a result of moving objects in the environment during different time periods.
Gonzales et al. [14]	Uses available/neighboring SSIDs and their average RSSI values as observed by a given wireless node to establish expected/normal environment for initiating connection with a given access point.	A valid approach for verifying the validity of an SSID before connecting a mobile wireless node to it. However, this approach cannot guarantee the absence of spoofing once the connection is established and is not useful in settings where no other SSID is available in the environment.

We have also evaluated the classification performance of the three models against an adversarial traffic mix (as explained in Section 5.2). We can observe in Table 2 (also refer to Figures 11 and 12, that our approach slightly loses classification accuracy (by 1%) when 20% of RSSI values in a given window is generated by an adversary while the performance of the other two classifiers deteriorates significantly. This can partly be explained by the fact that our LSTM (Long Short-Term Memory) autoencoder approach takes into consideration the order in which RSSI samples appear (i.e., are collected), while the other two approaches treat RSSI values in a window as independent data points. It is clear from the obtained results that our approach is well equipped to deal with an active adversary that transmits during the transmission period of the legitimate node while such overlap of traffic is not well protected using existing approaches.

Table 2. Passive Adversary, who assumes a single spot in the environment and does not adjust its transmission power.

		0% Mixed Window Content						**20% Mixed Window Content**						**50% Mixed Window Content**					
		Day Classifier			**Night Classifier**			**Day Classifier**			**Night Classifier**			**Day Classifier**			**Night Classifier**		
		Precision	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
Multi Model	Normal	1.0	0.95	0.97	1.0	0.99	0.99	1.0	0.93	0.97	1.0	0.99	0.99	1.0	0.93	0.97	1.0	0.99	0.99
LSTM Autoencoder*	Spoofed	0.97	1.0	0.98	0.99	1.0	0.99	0.93	1.0	0.96	0.98	1.0	0.99	0.93	1.0	0.96	0.98	1.0	0.99
One-Class	Normal	0.66	0.52	0.58	0.73	0.42	0.53	0.56	0.52	0.54	0.60	0.48	0.53	0.58	0.52	0.55	0.59	0.48	0.53
SVM [23] (baseline)	Spoofed	0.69	0.80	0.74	0.50	0.79	0.61	0.48	0.52	0.50	0.37	0.49	0.42	0.50	0.56	0.53	0.36	0.47	0.41
Log-likelihood	Normal	0.85	0.92	0.88	0.83	0.89	0.86	0.75	0.89	0.81	0.73	0.78	0.75	0.77	0.91	0.83	0.81	0.89	0.85
ratio [13]	Spoofed	0.87	0.90	0.88	0.92	0.95	0.93	0.76	0.81	0.78	0.84	0.83	0.83	0.80	0.83	0.81	0.85	0.92	0.88

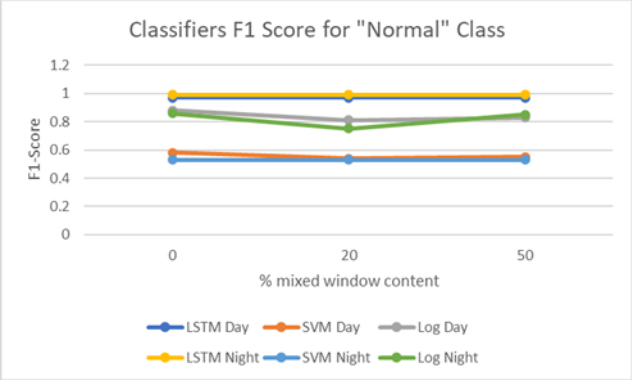

Figure 11. Comparison of 'Normal Classification' of our novel detection method with two other [13,23] state-of-the-art approaches proposed in the literature.

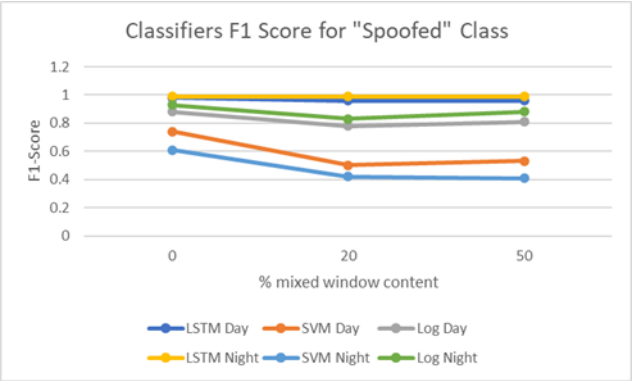

Figure 12. Comparison of 'Spoofed Classification' of our novel detection method with two other [13,23] state-of-the-art approaches proposed in the literature.

5.4. Model Switching at Runtime

In Section 4.2 we have explained the need for a bi-modal LSTM autoencoder classifier, and we have proposed a fully automated and adaptive approach to switching between the two train models/classifiers at runtime. Using the collected real-world datasets we have put this idea to test by continuously monitoring the reconstruction error of the two train models at runtime. As depicted in Figure 13, at night when the RSSI stream had relatively lower volatility, the night model (the blue line) resulted in low reconstruction error while the day model (the orange line) resulted in high reconstruction error—as to be expected. However, at the point in time when the volatility was about to pick up, we can observe a sudden jump in the night model's reconstruction error accompanied by significant improvement in the day model's reconstruction error, ultimately resulting in a crossover between the two error lines (orange and blue). This is a clear indication that the night model could be retired, and the day model could be activated for detection. Clearly, this demonstrates the viability of the *c*rossover indicator to facilitate an automated and adaptive switching schedule between the two trained LSTM autoencoder models.

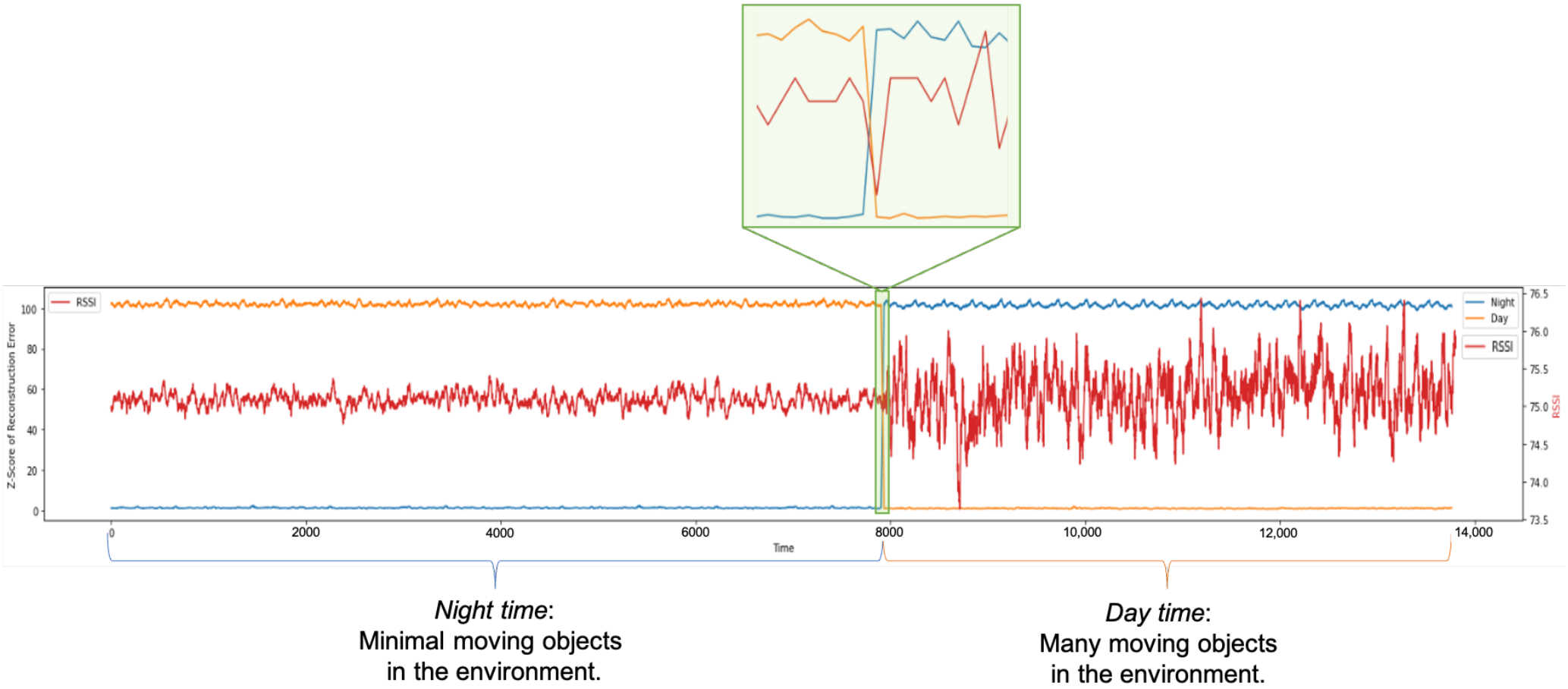

Figure 13. Tracking reconstruction error of two trained models during an entire day. The crossover point between the two reconstruction error lines (orange and blue) coincide with increase in volatility of RSSI stream (the red line)—a clear indicator to be used to switch between trained models.

6. Discussions and Conclusions

In this work we have proposed a novel RSSI-based MAC spoofing detection approach using a multi model LSTM autoencoder classifier. The advantages of our approach over earlier works in this field are twofold. First, our approach is capable of coping with periodic environmental (i.e., signal) disturbances caused by moving objects. Second, our approach can tolerate and detect presence of an adversary that transmits in close time intervals to legitimate network devices.

As part of this research, we have also studied the variability of RSSI streams in a real-world residential area, and (from the collected measurements) we have confirmed the existence of two very distinct periods in the observed RSSI streams (i.e., day vs night). These observations provide real-world justification for the use of a bi-modal LSTM autoencoder, with one autoencoder being trained for each variability period. In addition, we have proposed an automated and adaptive technique for determining the optimal point in time to switch between the two train models.

It may be worth clarifying that one of the key assumptions of our work is that the IoT network utilizing our solution is composed of a large number of sensing nodes (which are in charge of collecting and transmitting sensory readings from their immediate environment) and one or a few sink nodes (which are in charge of receiving and/or aggregating the sensory readings received from multiple sensor nodes). Furthermore, we assume that the sink nodes are generally more powerful (e.g., have better energy and processing capacity) compared to the sensing nodes.

Now, given the inherently 'one-way' nature of the assumed application and the respective communication patterns (i.e., sensors transmit while sinks receive), the most likely targets of an adversary existing in this environment (i.e., most likely recipients of spoofed packets) would be the sink nodes, and very rarely the 'ordinary' sensing nodes. Consequently, it is reasonable to assume that the proposed solution would have to be primarily, if not exclusively, implemented on the sink nodes in order help verify the authenticity of received sensor readings. As previously clarified, sink nodes are generally assumed to have reasonable energy and processing capabilities.

It is also worth pointing out that our proposed LSTM autoencoder approach is utilizing one-dimensional data (i.e., RSSI readings) as inputs, which makes the training of our model(s) extremely energy-inexpensive and fast, even for Zigbee IoT nodes as used in our experiments. Furthermore, using the trained LSTM autoencoders at runtime relies on very simple matrix multiplications, which are of similar complexity to SVM, linear regressions, or Gaussian models previously proposed in the literature, and which are well within the capabilities, even of IoT nodes, with limited energy and computational characteristics.

Given that most IoT networks have multiple participants, it is natural to wonder how our proposed method could be further expanded should participating nodes be capable and/or willing to cooperate with each other in order to detect an ongoing MAC spoofing attack. Although such an idea could likely enhance the overall detection and network performance, it also requires careful consideration and engineering in order to ensure robustness against (e.g.) potential byzantine nodes. We are planning an in-depth investigation of such a cooperative multi-node approach as one of the future research directions of our work.

In our previous work [10], we proposed an RSSI-based randomization technique for protection against an active adversary capable of modifying its transmission power and its location in the target/victim environment. Of course, such randomization could positively affect our novel proposed method but the classification performance might change drastically under a randomized schema. Finally, in this work we have assumed that the system operator is in charge of detecting low vs high volatility periods in the training RSSI time-series and divided the training set into two subsets for training the proposed bi-modal LSTM autoencoders. However, one could argue that due to variability in RSSI during the presence vs absence of moving objects, it is possible to detect two periods (for separating the training datasets for building the multi-model classifiers) using

unsupervised clustering approaches such as k-means instead of relying on the judgment of a system operator for creating such separation. This is certainly an interesting future work that can further enhance our proposed *crossover model switching indicator*.

Author Contributions: Conceptualization, P.M. and N.V.; methodology, P.M.; software, P.M.; validation, P.M. and N.V.; data curation, P.M.; writing—original draft preparation, P.M.; writing—review and editing, P.M. and N.V.; supervision, N.V.; project administration, N.V. Both authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lashkari, A.H.; Danesh, M.M.S.; Samadi, B. A survey on wireless security protocols (WEP, WPA and WPA2/802.11 i). In Proceedings of the 2009 2nd IEEE International Conference on Computer Science and Information Technology, Beijing, China, 8–11 August 2009; pp. 48–52.
2. The Independent IT Security Institute AX Test. 2017. Available online: https://www.iot-tests.org/2017/06/hue-let-there-be-light/ (accessed on 1 September 2020).
3. Wood, A.D.; Stankovic, J.A. Denial of service in sensor networks. *Computer* **2002**, *35*, 54–62. [CrossRef]
4. Nguyen, V.L.; Lin, P.C.; Hwang, R.H. Energy depletion attacks in low power wireless networks. *IEEE Access* **2019**, *7*, 51915–51932. [CrossRef]
5. Ahmad, M.S.; Tadakamadla, S. Short paper: Security evaluation of IEEE 802.11 w specification. In Proceedings of the Fourth ACM Conference on Wireless Network Security, Hamburg, Germany, 14–17 June 2011; pp. 53–58.
6. Wu, W.; Gu, X.; Dong, K.; Shi, X.; Yang, M. PRAPD: A novel received signal strength-based approach for practical rogue access point detection. *Int. J. Distrib. Sens. Netw.* **2018**, *14*, 1550147718795838. [CrossRef]
7. Moosavirad, S.M.; Kabiri, P.; Mahini, H. RSSAT: A Wireless Intrusion Detection System Based on Received Signal Strength Acceptance Test. *J. Adv. Comput. Res.* **2013**, *4*, 65–80.
8. Demirbas, M.; Song, Y. An RSSI-based scheme for sybil attack detection in wireless sensor networks. In Proceedings of the 2006 International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'06), Buffalo-Niagara Falls, NY, USA , 26–29 June 2006; p. 5.
9. Aghabozorgi, S.; Shirkhorshidi, A.S.; Wah, T.Y. Time-series clustering—A decade review. *Inf. Syst.* **2015**, *53*, 16–38. [CrossRef]
10. Madani, P.; Vlajic, N.; Sadeghpour, S. MAC-Layer Spoofing Detection and Prevention in IoT Systems: Randomized Moving Target Approach. In Proceedings of the 2020 Joint Workshop on CPS & IoT Security and Privacy, Lisbon, Portugal, 15 September 2020; pp. 71–80.
11. Faria, D.B.; Cheriton, D.R. Detecting identity-based attacks in wireless networks using signalprints. In *Proceedings of the 5th ACM Workshop on Wireless Security*; ACM: New York, NY, USA, 2006; pp. 43–52.
12. Chen, Y.; Trappe, W.; Martin, R.P. Detecting and localizing wireless spoofing attacks. In Proceedings of the 2007 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, San Diego, CA, USA, 18–21 June 2007; pp. 193–202.
13. Sheng, Y.; Tan, K.; Chen, G.; Kotz, D.; Campbell, A. Detecting 802.11 MAC layer spoofing using received signal strength. In Proceedings of the IEEE INFOCOM 2008—The 27th Conference on Computer Communications, Phoenix, AZ, USA, 13–18 April 2008; pp. 1768–1776.
14. Gonzales, H.; Bauer, K.; Lindqvist, J.; McCoy, D.; Sicker, D. Practical defenses for evil twin attacks in 802.11. In Proceedings of the 2010 IEEE Global Telecommunications Conference (GLOBECOM 2010), Miami, FL, USA, 6–10 December 2010; pp. 1–6.
15. Nelson, B.; Rubinstein, B.I.; Huang, L.; Joseph, A.D.; Lee, S.J.; Rao, S.; Tygar, J. Query Strategies for Evading Convex-Inducing Classifiers. *J. Mach. Learn. Res.* **2012**, *13*, 13–23.
16. Madani, P.; Vlajic, N. Robustness of deep autoencoder in intrusion detection under adversarial contamination. In Proceedings of the 5th Annual Symposium and Bootcamp on Hot Topics in the Science of Security, Raleigh, NC, USA, 10–11 April 2018; pp. 1–8.
17. Goodfellow, I.; Bengio, Y.; Courville, A.; Bengio, Y. *Deep Learning*; MIT press: Cambridge, UK, 2016; Volume 1.
18. Kim, J.; Kim, J.; Thu, H.L.T.; Kim, H. Long short term memory recurrent neural network classifier for intrusion detection. In Proceedings of the 2016 International Conference on Platform Technology and Service (PlatCon), Jeju, Korea, 15–17 February 2016; pp. 1–5.
19. Luo, W.; Liu, W.; Gao, S. Remembering history with convolutional lstm for anomaly detection. In Proceedings of the 2017 IEEE International Conference on Multimedia and Expo (ICME), Hong Kong, China, 10–14 July 2017; pp. 439–444.

20. Malhotra, P.; Ramakrishnan, A.; Anand, G.; Vig, L.; Agarwal, P.; Shroff, G. LSTM-based encoder-decoder for multi-sensor anomaly detection. *arXiv* **2016**, arXiv:1607.00148.
21. Safaric, S.; Malaric, K. ZigBee wireless standard. In Proceedings of the ELMAR 2006, Zadar, Croatia, 7–9 June 2006; pp. 259–262.
22. Zhang, Z. Improved adam optimizer for deep neural networks. In Proceedings of the 2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS), Banff, AB, Canada, 4–6 June 2018; pp. 1–2.
23. Laxhammar, R. Conformal Anomaly Detection: Detecting Abnormal Trajectories in Surveillance Applications. Ph.D. Thesis, University of Skövde, Skövde, Sweden, 2014.