Wi-Fi attack classification with Deep Learning

Vittorio Triassi
vittorit@stud.ntnu.no
TTM4137 Wireless Security Technical Essay

October 28, 2019

1 Introduction

Since IEEE 802.11 was introduced [1], several have been the concerns about the ease through which malicious attacks could be performed. Nowadays, everything is getting more and more connected [4], and that is the reason why it has become of particular importance to be able to correctly detect such attacks, in order to distinguish them from the regular traffic. Our goal in this essay is to discuss about ways to classify different types of attacks in 802.11 by using Deep Learning tecniques [10]. The reason why it gets interesting to resort to Deep Learning, finds its roots in the abundance of data we have at our disposal today, which results in an emergence of several new applications. From now on, the essay will be divided into a *discussion* part, in which we will talk about attack detection, then we will move to Deep Learning tecniques and finally *conclusions* will be drawn, getting insights on the problem addressed.

2 Discussion

2.1 Attack detection

When interested in performing attack detection and classification tasks, we can typically use two approaches: 1) signature-based and 2) anomaly-based detection [10, 12]. In a *signature-based* approach we look for specific patterns that identify the malicious attacks and compare the incoming traffic with the signatures we have. If we want to classify novel attacks, this approach might not be very advisable. On the other hand, in an *anomaly-based* approach, we take advantage of Machine Learning models that are trained on specific data. If the incoming traffic deviates from the usual one, we detect an anomalous behaviour. Since we want to classify also novel attacks, the latter approach is to be preferred. It is worth to mention that our goal here is not to just categorize the traffic in "regular" and "malicious" as if it was a binary classification. We want to address the

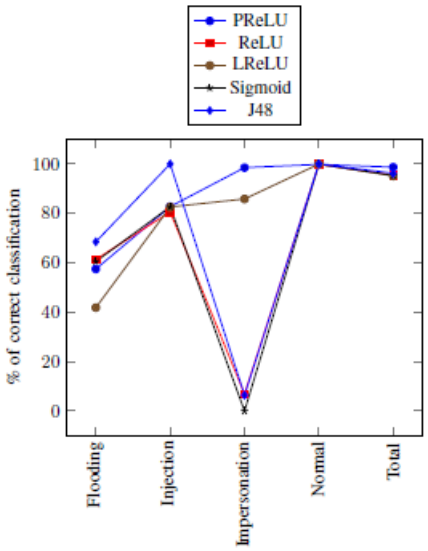

Figure 1: Results obtained with the SAE architecture [10].

problem as a multi-class classification task instead [9]. More specifically, we are interested in classifying: (a) injection attacks: the attacker sends a lot of small and correctly encrypted data frames over the network, (b) flooding attacks: the aggressor generates a huge volume of frames in a very short time and (c) impersonation attacks: the attacker takes over the identity of one of the parties [5, 10].

2.2 Deep Learning tecniques

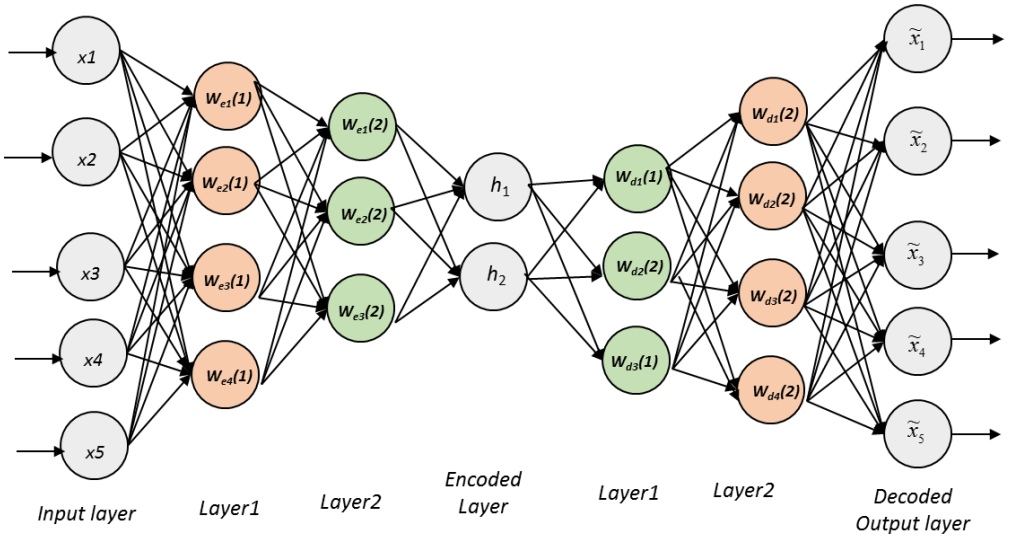

Figure 2: Stacked Auto-Encoder (SAE) [8].

Several are the classification algorithms that can be used in Machine Learning. Some of them are linear classifiers such as Logistic Regression and the Naïve Bayes classifier or others like Support Vector Machines, Decision Trees, Random Forest and Neural Networks. For the sake of brevity, in our discussion we will involve just Neural Networks. In [10], a Deep Learning approach is proposed to address the problem of attack classification. In the cited work, a Stacked Auto-Encoder (SAE) was used. A SAE is a Neural Network made up of several layers in which the output of each layer is connected to the input of the following one, and so on. The idea behind a SAE is to take as input a high-dimensional vector and compress it into a smaller space. The encoded layer is also known as "bottleneck". From here, we want to reconstruct the original input (see Figure 2). To understand how such computation is performed, let us consider W^l and b^l as the parameters required for the l^{th} auto-encoder and let $a^{(l)}$ be the activation that takes as input z^l. Each layer would be encoded as:

$$a^{(l)} = f(z^{(l)}) \tag{1}$$

where $f(z^{(l)})$ is the activation function.

$$z^{(l+1)} = W^{(l)}a^{(l)} + b^{(l)} \tag{2}$$

In order to perform the decoding instead, we simply perform the computation going backwards through the layers. The reason why such approach is interesting is that it is able to self-learn the features necessary to correctly discriminate an attack class from the others once the model is trained. To train the model, we use a dataset [7] that is split into training data and test data. Each training example represents a single packet in the traffic. Such packets are stored as vectors of 155 attributes and provide several information, among which there are information on the MAC layer (IV, source and destination address, etc).

Tools & Hardware To perform the attacks, the authors in [5], set up a lab in which a few mobile devices were considered as legitimate clients in the network. An AP was used and this implemented the WEP encryption. To carry out the experiment, an attack node was used to launch 15 different attacks. The tools utilized were the Aircrack suite [2], the MDK3 tool [11] and the Metasploit framework [6]. All the attacks that were carried out by following a similar process, have been categorized into one of the three classes mentioned at the end of Section 2.1.

Experimental Results Once we have defined the model that will be used and how the attacks are performed, it is time to understand what results we obtain for the aforementioned classes. A couple of different architectures were used, and they were respectively SAEs made up of 2 and 3 hidden layers. In Figure 1, the results with the 2-hidden-layer model are shown. In this model is possible to appreciate, among the other activation functions, the very good results we obtain when using the PReLU [3]. Such model achieves an overall accuracy of 98.6% that is strongly related to the ability of classifying the impersonation attacks. Another architecture was tried and it consisted of 3 hidden-layers. In this case, a drop in the performance for the flooding attacks was noticed. The reason seems to be related to the inability of the model to learn good features for such attacks. Overall, both architectures seem to be quite robust to address our problem and the 2-hidden-layer model was able to perform better than the deeper one. The interesting aspect that is worth pointing out is that when performing the 15 attacks, some of them did not appear during the training phase, and our model was able to achieve a good accuracy also on novel attacks, which was the main goal of our discussion.

3 Conclusion

In the following essay, a Deep Learning approach to classify Wi-Fi attacks was presented. The basic idea was to combine the good performance that Machine Learning algorithms can achieve to address a more general problem such as anomaly detection in a Wi-Fi architecture. It seems quite reasonable to think that the approach discussed here can be easily extended to several other applications in information security. For the purpose of our discussion, it was interesting to provide an additional diagnostic tool that falls into a trending field like Deep Learning. Another reason that brought this topic up is related to the huge amount of data we have access to nowadays. Thanks to these new approaches we can think to automate processes that should be performed manually otherwise. In our discussion, only Neural Networks, and more specifically Stacked Auto-Encoders were considered. In [5], also experiments with Random Forest were conducted and these achieved really high performance. Moreover, despite in our discussion the encryption mechanism was provided by WEP, it is possible to use the same dataset for WPA and WPA2 as well. Similar tests can be carried out also with different wireless technologies such as WiMAX and LTE since they share some weaknesses we have already exploited.

References

[1] IEEE 802.11. Ieee 802.11 wireless local area networks, the working group for wlan standards. http://www.ieee802.org/11/. Accessed: 2019-10-24.

[2] Aircrack-ng. Wi-fi network security tools. https://www.aircrack-ng.org/. Accessed: 2019-10-27.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[4] Qi Jing, Athanasios V Vasilakos, Jiafu Wan, Jingwei Lu, and Dechao Qiu. Security of the internet of things: perspectives and challenges. *Wireless Networks*, 20(8):2481–2501, 2014.

[5] Constantinos Kolias, Georgios Kambourakis, Angelos Stavrou, and Stefanos Gritzalis. Intrusion detection in 802.11 networks: empirical evaluation of threats and a public dataset. *IEEE Communications Surveys & Tutorials*, 18(1):184–208, 2015.

[6] Metasploit. Penetration testing software. https://www.metasploit.com/. Accessed: 2019-10-27.

[7] University of the Aegean. Awid - wireless security datasets project. http://icsdweb.aegean.gr/awid/features.html. Accessed: 2019-10-27.

[8] Packtpub.com. Setting up stacked autoencoders - r deep learning cookbook. https://subscription.packtpub.com/book/big_data_and_business_intelligence/9781787121089/4/ch04lvl1sec51/setting-up-stacked-autoencoders. Accessed: 2019-10-26.

[9] Masashi Sugiyama. *Introduction to statistical machine learning*. Morgan Kaufmann, 2015.

[10] Vrizlynn LL Thing. Ieee 802.11 network anomaly detection and attack classification: A deep learning approach. In *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6. IEEE, 2017.

[11] Kali Tools. Mdk3. https://tools.kali.org/wireless-attacks/mdk3. Accessed: 2019-10-27.

[12] Wikipedia. Intrusion detection system. https://en.wikipedia.org/wiki/Intrusion_detection_system#Signature-based. Accessed: 2019-10-24.