Intrusion Anomaly Detection and Prevention in 802.11 Networks : A Deep Learning Approach

« An evil twin detection Cybersecurity Attack on the user's side »

Keywords: IoT security; intrusion detection system; Deep learning; Transfert Learning

Update 25.02.2022
(OPTIMIZATION)

Introduction (see the previous slides)

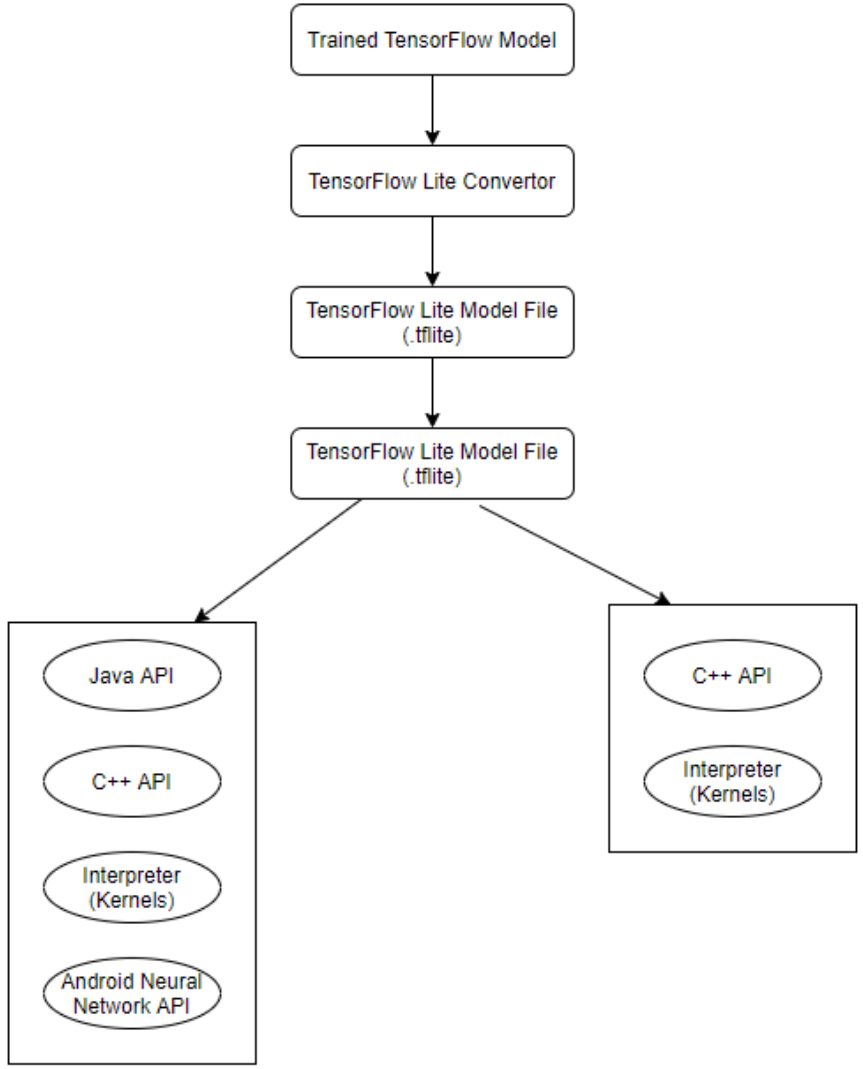
- IoT devices are connected usually over wireless networks, eavesdropping can be used to access the private information of a communication channel [10,11].

- Deep Learning (DL) based techniques have recently gained credibility in a successful application for detecting network attacks, including IoT networks. But it still poses challenges in terms of detection in an unknown environment and the type of attack to be detected.

- Network traffic can be captured and studied to learn normal patterns. Any deviation from these learned normal patterns can be used to detect anomalous behavior.

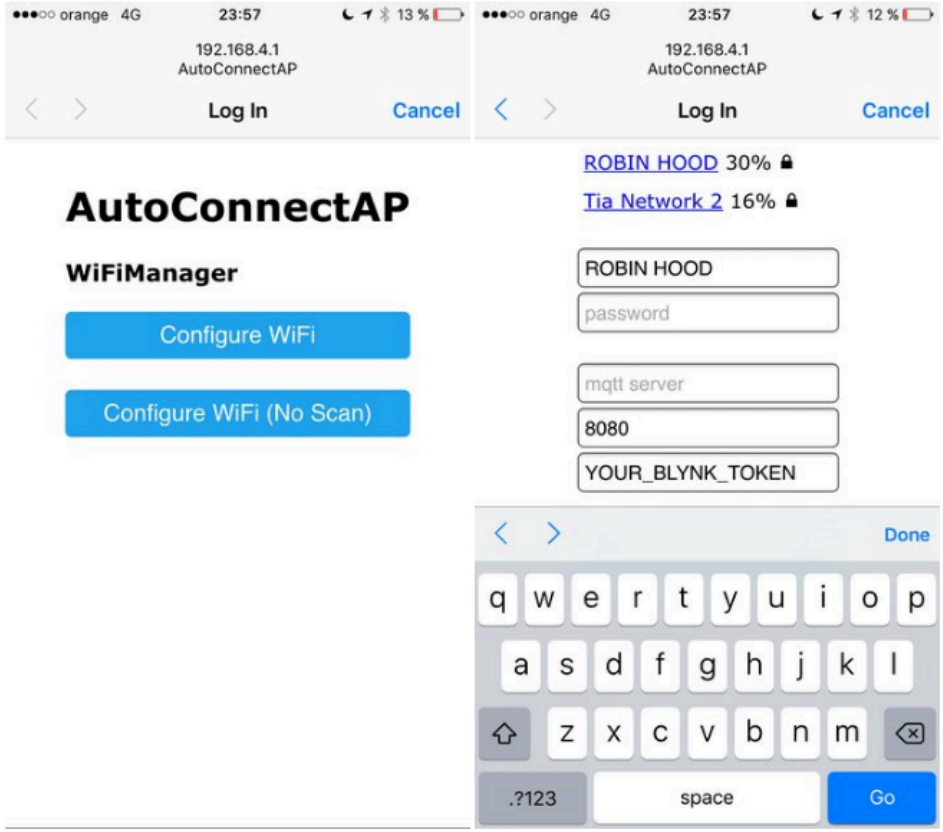
Existing work using a Lite deep learning framework

1. Optimization of computation time during transfer learning
2. The ability to train a model and then convert it to a lite model (compact and not too heavy)
3. Research articles have already worked on methods to facilitate the computation of each model on devices with low CPU capacity.
4. Tensorflow offers a d-structure to convert a pre-trained model into a lite model → This is what we will use to optimize our computing time.



At the level of access to wifi and model

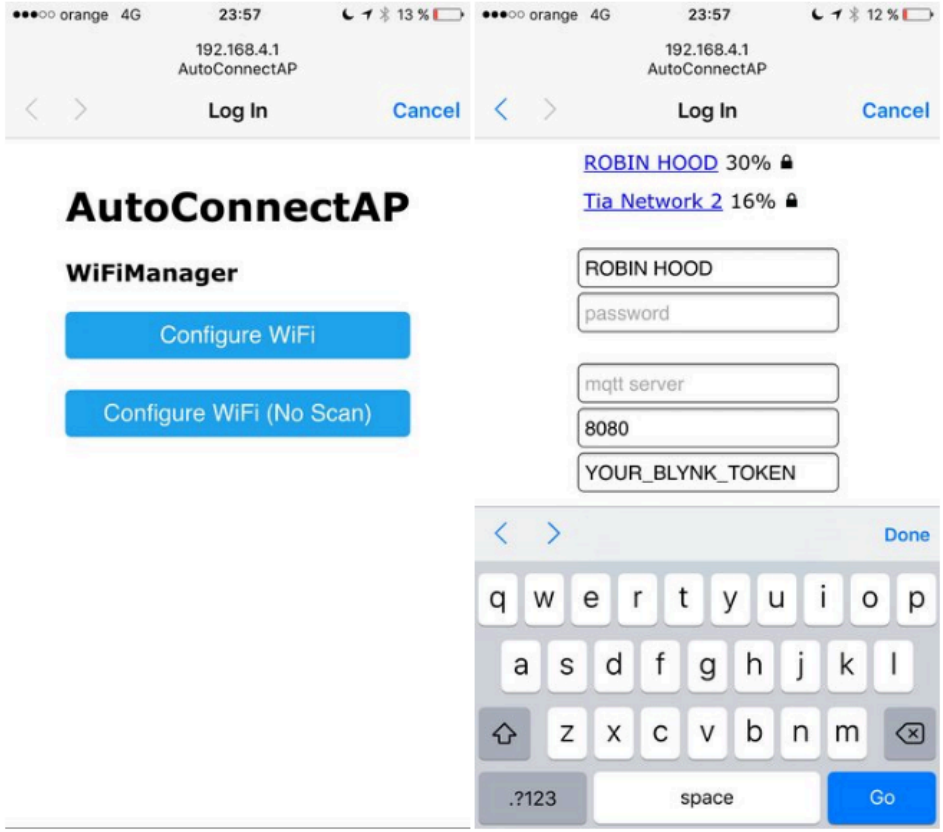
1. We also tried to create a window to call our attack detection model!
2. This one is still on a local server for the moment
3. maybe develop an application to take the pre-trained model and then do the learning by transfer
4. Demo will be possible with 3 ESP32 cards (one is to attack and one as a legitimate AP and the last one to implement by our model with the ESP server).



At the level of access to wifi and model

1. We also tried to create a window to call our attack detection model!
2. This one is still on a local server for the moment
3. maybe develop an application to take the pre-trained model and then do the learning by transfer
4. demo will be possible with 3 ESP32 cards (one is to attack and one as a legitimate AP and the last one to implement by our model with the ESP server).

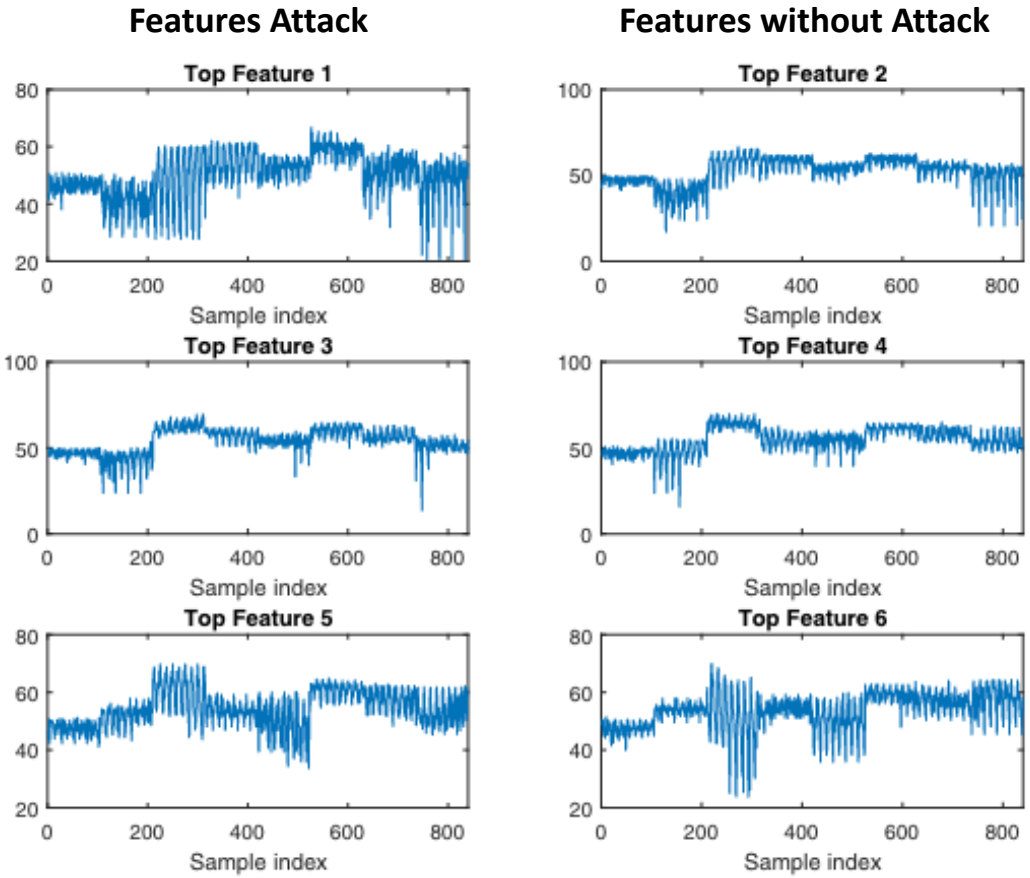
→ But I still have to fix some bugs in the wifi scan so that it is automatic!!



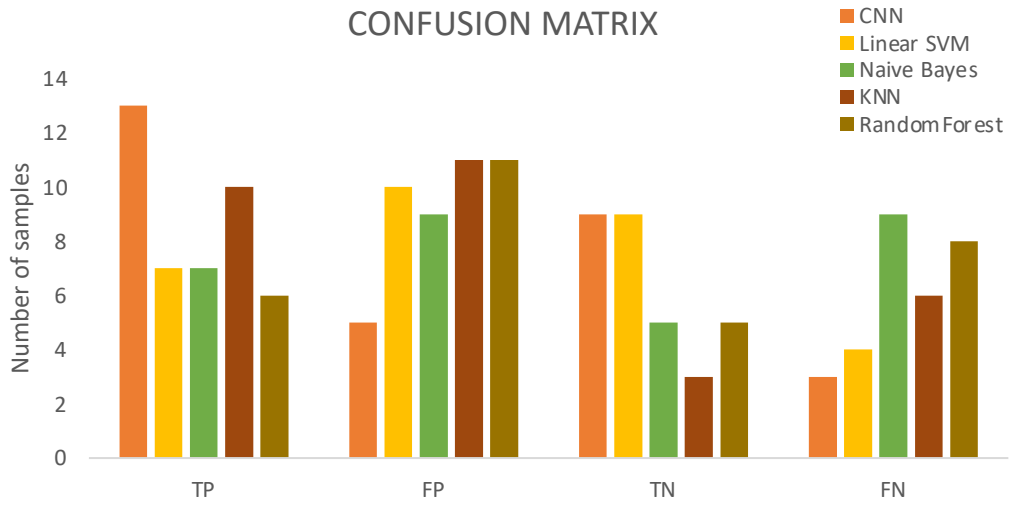
Generating Data (positive and negative samples)

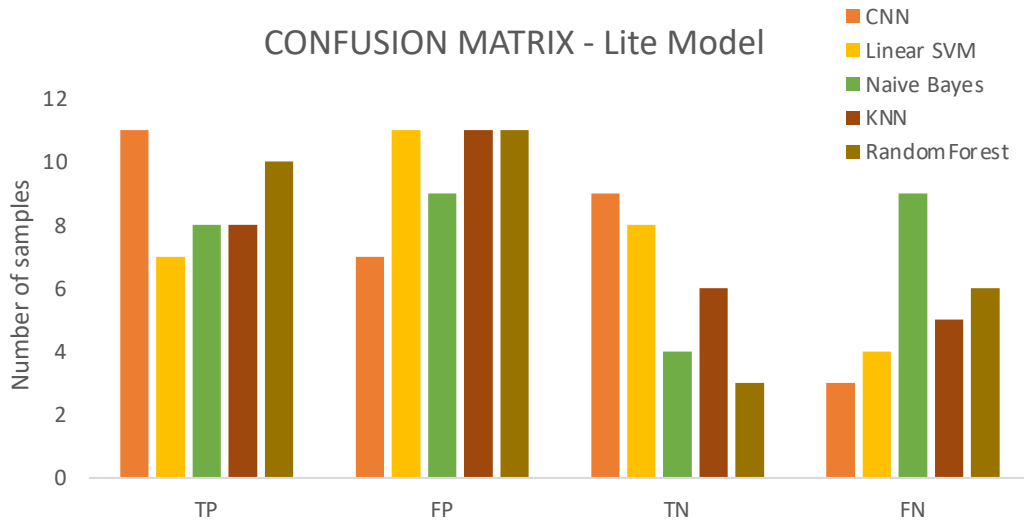
- Concerning the features we tried to optimize the features of CSI, but it seems to me that it is not obvious to make the separation with what is useful and what is not!

- Here for example (1-2), (3-4) features for attack and without attack there are only the first features which are different but the rest it is almost the same.



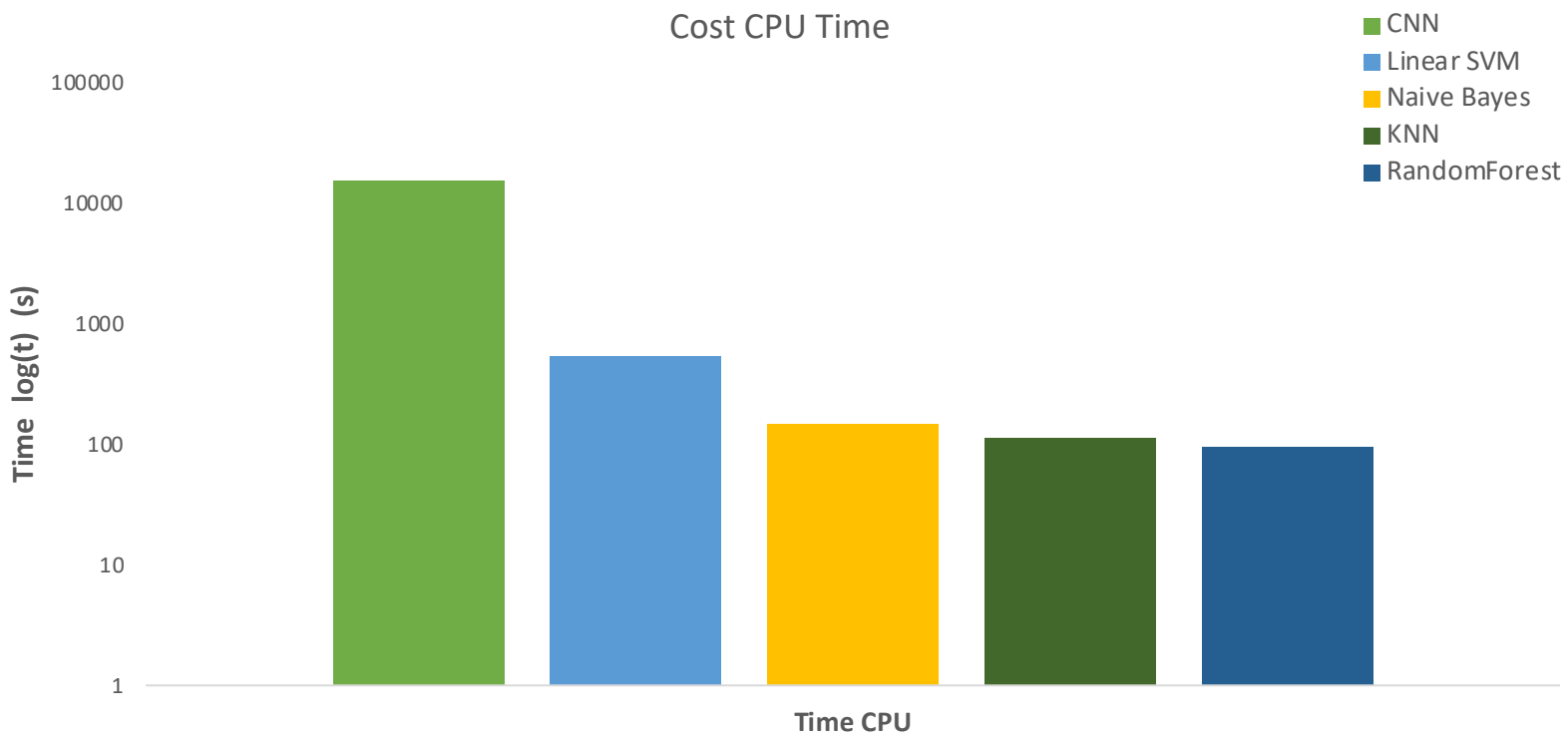
Transfer learning in a general scenario – Result





We had to stop the training at the time 310 epoch because it took more than 7h46 hours → Time complexity

CPU Time



Random forest gives surprising results!!

Remarks

- One of the major problems we have is the lack of features, we have to find a way to increase the number of features in question.
- Capturing the signal during the attack, we don't know if it's the right method, because it can be a passive attack.
- We are trying to build an autoencoder in transfer learning format, which will be a great advantage over our current framework, because the reconstruction of the signal itself over different periods of time could help us to find Evil-Twin APs.