Technical Disclosure Commons

Defensive Publications Series

November 2019

WIRELESS TRAFFIC ANALYSIS AND ANOMALY DETECTION USING DEEP LEARNING

Yogesh Kondareddy

Sai Dantu

Ramadhasan Thangachamy

Santosh Kulkarni

Leela V Kiran Kumar Chirala

See next page for additional authors

Inventor(s)
Yogesh Kondareddy, Sai Dantu, Ramadhasan Thangachamy, Santosh Kulkarni, Leela V Kiran Kumar Chirala, and Rahul Singh

WIRELESS TRAFFIC ANALYSIS AND ANOMALY DETECTION USING DEEP LEARNING

AUTHORS:

Yogesh Kondareddy
Sai Dantu
Ramadhasan Thangachamy
Santosh Kulkarni
Leela V Kiran Kumar Chirala
Rahul Singh

ABSTRACT

Presented herein are innovative techniques for analyzing network traffic and identifying anomalous patterns using Artificial Intelligence (AI). In particular, the techniques presented herein map the network traffic into pictures and use advanced image recognition AI to detect anomalies in those pictures. The solution uses Transfer Learning from pre-trained models such as RESNET50. Since the models are pre-trained, the amount of new training data and time is reduced drastically.

DETAILED DESCRIPTION

Wireless (IEEE 802.11) network traffic analysis and anomaly detection has been well studied, with most of the existing solutions being rule-based or deep packet inspection mechanisms. The problem with these approaches is that new attack signatures can still sneak in. The use of Artificial Intelligence (AI) for traffic analysis is becoming very popular in the networking industry and has been showing promising results, but AI faces fundamental challenges associated with labeling huge amounts of data, long training times, and expensive integration requirements. The techniques presented herein include an innovative way to analyze network traffic using transfer learning in AI while addresses all the above-mentioned challenges.

Transfer learning is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task. The techniques presented herein map a stream of packets into pictures and then use existing pre-trained image recognition models, such as RESNET50, to analyze the pictures and detect anomalous patterns. The

techniques presented herein are primarily described with reference to application to wireless traffic. However, it is to be appreciated that the techniques presented herein may also be applicable to wired traffic.

It is known that complex problems can be solved with AI and, image recognition techniques, in particular, have successfully applied AI to a number of problems. Moreover, there are open-source Image recognition models, such as RESNET50, which have been pre-trained with millions of images. These existing models can be used for recognizing anything with very little training.

However, such pre-trained models do not exist for networking uses and the techniques presented herein may utilize new Deep Learning Neural Network (DPNN). However, numerous amounts of categorized data is needed to train such DPNNs. To address this issue, the techniques presented herein transfer the existing image classification knowledge to networking traffic so as to use DPNN for traffic analysis with little data. To implement such techniques, several steps are performed in the training phase. These steps include:

1. Simulate a potential anomaly
2. Capture wireless packets
3. Extract relevant information (Pre-processing)
4. Create Images (Pre-processing)
5. Train a Pre-trained model such as RESNET
6. Obtain the trained model

This process has been demonstrated for one simple use case, namely client connectivity issues, with very good success. Figures 1, below, depicts the steps where some fields of the packets are extracted and images are created using those fields. The image in Figure 1 is a good case where the phone was able to connect successfully. However, the second image in Figure 2, below, is an anomalous case where an incorrect password was used and the connection failed. These images were used in training RESNET50 with initial weights from Imagenet.

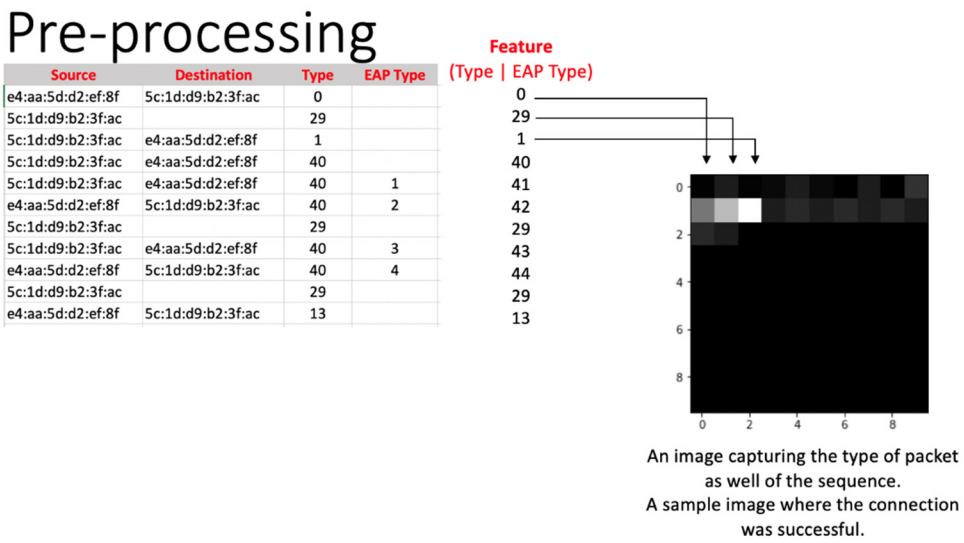

Figure 1

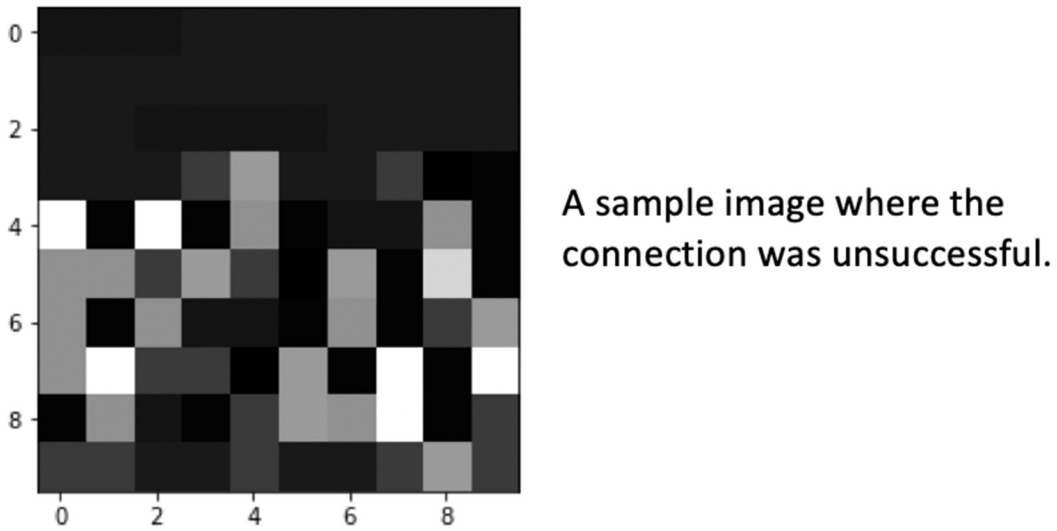

Figure 2

Once the training phase is performed, the detection phase includes the following steps:

1. Capture real-time wireless traffic
2. Extract relevant information
3. Create Images
4. Run it through the trained model
5. Classified output

Image Creation:

A protocol is a sequence of defined packet formats (i.e., a pattern). The techniques presented herein propose to teach the neural network the protocol patterns and the ability to categorize the patterns. When the packets are encoded into an image, the order of pixels inherently captures the sequence of the packets. As long as all the necessary information from the packets is encoded into the image, a Deep Neural Network should be able to learn the packet sequence and be able to categorize them. So, the key is in the way the packets are encoded into an image. The image creation process is also a unique element of the techniques presented herein.

In general, each image represents the packets of a single client. Once the packets are captured, the data is filtered per MAC address and one image is created for every specified set of packets of that MAC address. Creating each image using the packets of a single MAC address makes it easy to capture the behavior of a specific device and avoids randomness due to the presence of other devices. The following parameters are defined to describe the image creation process:

- **Packet Stream:** A set packets relating to a single Client/MAC address in the MAC header.
- **Packcell**: The group of pixels depicting a single packet in an image.
- **Pixels Per Packet (PPP):** Number of pixels used to depict the information of each packet. This translates to the number of bytes encoded for each packet.
- **Packet Per Image (PPI):** Number of packets depicted in an image.
- **Scaling Factor (SF):** Number of times each Packcell is replicated.
- **Raw Encoding:** Raw packet data is encoded into the Packcell.
- **Mined Encoding:** Filtered information from the packet is encoded into the Packcell.

Figure 3, below, describes various parameters of the image conversion process, where, for simplicity, only one of the RGB layers is shown.

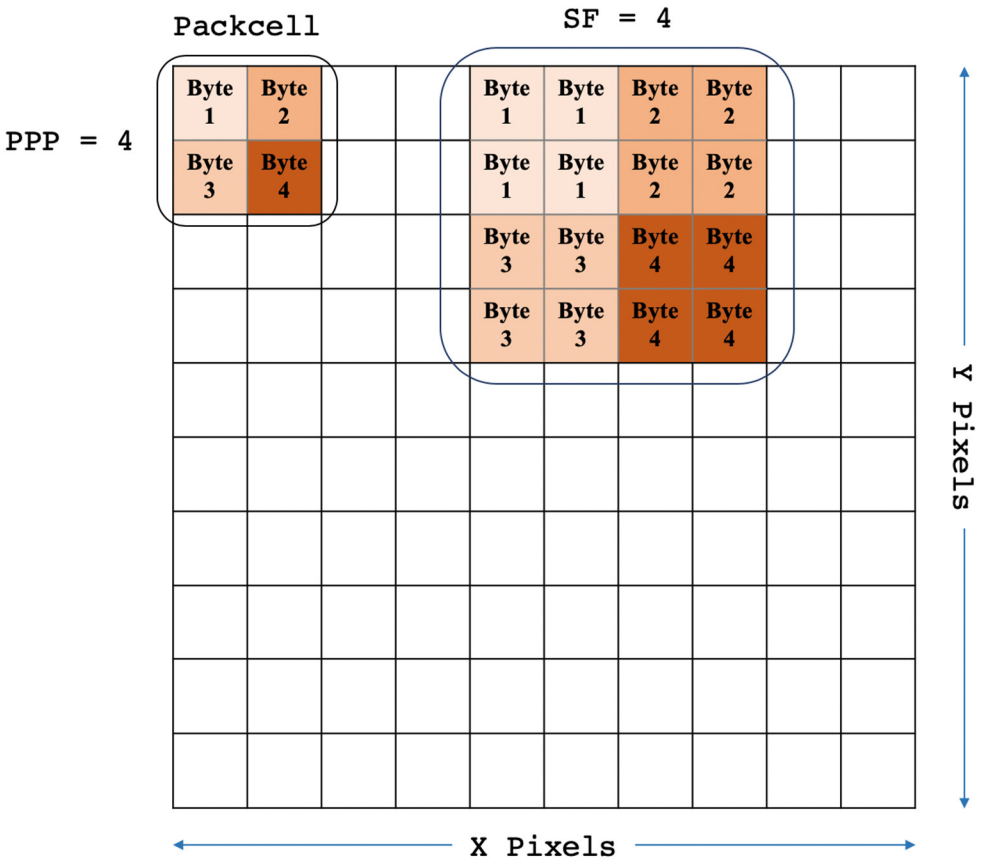

Figure describing various parameters of the image conversion process. Only one of the RGB layers is shown for simplicity.

Figure 3

Given a set of wireless packet data either captured through sniffing tools or through the other infrastructure such as Access Points, the set of pixels are used to represent each packet. This set is called a Packetcell. Each pixel in an image is 1 Byte in size and, as such, the number of pixels in a Packetcell depends on the amount of information to encode from each packet. This parameter is defined as Pixels Per Packet (PPP). If PPP number of bytes is directly obtained from the packet and encoded, this is referred to as Raw encoding. If certain fields are picked from the packet and then encoded into the Packetcell, this is

referred to as Mined encoding. The advantage of Mined encoding is that less data is required to train the neural network, while taw encoding is a little less processor-intensive, but requires more labeled data to train covering variations in all fields of the packet. Each pixel in an image may need to be replicated sufficiently so not to loose information in Pooling operations.

In the techniques presented herein, Scaling Factor (SF) defines the amount of replication. Choosing a good SF is important so that the data in each packet is replicated sufficiently so as not to loose information in Pooling layers of the image recognition models. The optimal SF value needs to be found with some experimentation and other parameters may have to be adjusted accordingly.

Given the image dimensions and the required amount of information (bytes) to be encoded per packet, the above parameters can be adjusted to obtain desired results using the below formula:

$$(X * Y * Z) = (SF * PPP * PPI)$$

For example, RESNET50 operates on a 224x224x3 image by default. So, 150,528 Bytes can be encoded in one image. Suppose it is desirable to encode 100 packets per image (PPI) and the Scaling Factor is 15, then based on the above formula, PPI = 100. This means that it is possible to encode 100 bytes of information for each packet, which should be sufficient to include most of the headers of the packet in raw format.

Although the core idea of the techniques presented herein is to use pre-trained image recognition models, the same idea can be achieved by using other kinds of pre-trained models, such as by using speech recognition models. In such a case, a new approach to convert the packets to the required format may need to be defined. Stated differently, the proposed use of transfer learning for detecting network traffic can be applied in other contexts.

As noted above a protocol is a sequence of defined packet formats (i.e., a pattern) and the techniques presented herein teach a neural network the protocol patterns and the ability to categorize the patterns. When the packets are encoded into an image, the order of pixels inherently captures the sequence of the packets and, as long as all the necessary

information from the packets is encoded into the image, a Deep Neural Network should be able to learn the packet sequence and be able to categorize them.

In one demonstrated example, a network was trained with packet captures from a WLAN configured with PSK security. However, the system was able to identify failures on 802.1x WLAN, even if the sequence of EAP messages was different and there was an extra certificate exchange. In fact, the system was able to detect a failure even when there was no following de-authentication message or even if there were retries. This proved that the model actually learned the pattern and not just a discrete event.

Another example use of the techniques presented herein is a denial of service attack. A DoS attack can be launched using different types of frames on the wireless medium. It may be possible to train a neural network to learn that a repeating pattern is actually a DoS attack. If this is performed using scripts, the system may need to write some state machines to handle the patterns.

Advantages of the techniques presented herein include:

1. Need very little data: Since the image recognition models are pre-trained, little data is needed to curate the model to the specific needs.
2. Very little training time: Since the image recognition models are pre-trained, training them with data takes little time.
3. Minimal training infrastructure: Training can happen in the cloud.
4. Flexible installation infrastructure: The model can be deployed on a phone or the cloud to detect anomalies provided it has access to wireless traffic.
5. No privacy issues in collecting data. The techniques do not require collection of large data sets from customers in order to perform training, thus no privacy issues.

Demonstration:

The techniques presented herein have been used to train the RESNET50 model with less than 200 instances of successful and unsuccessful client connection sequences, while achieving 100% accuracy when testing on a wireless network. As such, the techniques presented herein have been demonstrated with client connection sequences, but this can be easily extended other anomalies, such as Denial of Service (DoS) attacks, throughput

issues, sequence number jumps, *etc*. Although association failure is not an anomaly, this use was selected to prove the concept that packet patterns and their sequence can be embedded into images and a Deep Neural Network can be trained to categorize them. As such, the techniques presented herein could also use to analyze the network traffic and not just for detection of anomalies.