MITM Detection Mechanism at client side using Machine Learning Algorithms

***Abstract*— Man-in-the-Middle attacks are talked about a lot in the technical field of computer security because they are one of the biggest concerns for professionals. MitM (Man in the Middle) attacks occur when an attacker intercepts data in the middle of a conversation by using a technique of interjecting themselves. Aside from attacking the data flow between the endpoints, attackers compromise the integrity and confidentiality of that data, as well. Through communication interception, an adversary can eavesdrop on confidential information and modify message integrity. Additionally, an adversary may intercept, modify, or destroy messages to end communication for one of the parties, thereby constituting a compromise of availability. In order to prevent the man in the middle attacks there are several methods, firewalls and intrusion detection systems, but unfortunately these methods are only applied to the administrative side of operations, i.e., Enterprise WIDS (Wireless Intrusion Detection System). The client system is not equipped to detect, prevent, and control these attacks, making it vulnerable to attack for the attacker. Our goal is to build a quality Intrusion Detection system using Anomaly Detection in Machine Learning that will detect network attacks on a client-side basis in order to prevent those attacks from happening.**

I. Introduction

We are dependent on Internet and cellular networks today to conduct virtually every aspect of our lives. For example, we conduct home banking online, enjoy online entertainment and shop, make use of social networks, etc. These services contain confidential data such as the banking information, personal information and other sensitive information which becomes key target to the attacker. Aside from the individuals, cybercriminals also attack the organizations and enterprises resulting in financial losses. We sometimes read about cyber-attacks on connected things and online services on a daily basis thanks to the Internet which keeps people and things always connected. A popular attack is Man-in-the-Middle (MITM), which allows hackers to gain control of users' data as it passes through the internet.

A man-in-the-middle (MITM) attack occurs when the attacker eavesdrops on communications between two targets, then secretly relays and possibly alters messages between those parties believing they're directly communicating. Attackers might be able to intercept data and information from legitimate parties while also sending malicious links or other information via the Internet in a way that is unlikely to be detected until it is too late.

A. How MITM works

The MITM attack generally consists of two phases: interception and decryption.

1) Phase 1: Interception: The interception phase of cybercrime involves hackers gaining entry to a network from an open or poorly secured Wi-Fi router, and/or through the use of DNS servers modified to manipulate address directives (DNS). A hacker will then test the router to find potential vulnerabilities and ways of gaining access. Usually that is achieved via a weak password, regardless of the fact that cyber criminals are also capable of doing more advanced things like IP spoofing or cache poisoning.

As soon as a target is determined, an attacker usually deploys data-stealing equipment to obtain access to and gather transmitted info from the victim, redirect traffic or otherwise control the user's web experience.

2) Phase 2: Decryption: Decryption is the second phase of an MITM attack. This is when stolen data is decoded and made accessible to cybercriminals. Decrypted data is used for nefarious purposes, such as identity theft, unauthorized purchases, and other fraudulent activities. There are cases where man-in-the-middle attacks are conducted for no apparent reason other than to disrupt enterprise operations and create chaos.

The demonstration of MITM attack can be shown in the below figure 1 where the attacker acts as a legitimated access point as HOME and intercepts all the traffic between the victim and the access point.

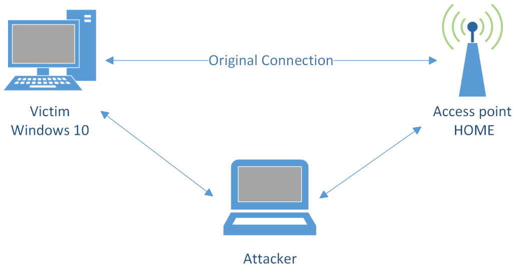


Fig. 1. MITM attack.

B. Types of MITM attacks

Depending on the target and the objective, there are a wide range of MITM techniques and outcomes. Below are some of the MITM discussed.

1) **DNS Spoofing:** Domain Name System (DNS) spoofing is a form of computer security hacking that involves the addition of false information to the DNS resolver cache and the use of altered DNS records to redirect online traffic to a fictitious website that looks like the destination website[1].
2) **IP Spoofing:** Cybercriminals use intellectual property spoofing to deceive users into thinking they are a part of a legitimate entity. The technique lets them gain access to computers within a target network, release sensitive information, launch DoS attacks, or compromise computer systems.
3) **HTTPS Spoofing:** An HTTPS spoof sends a false endorsement to the victim's browser once the request for a secured site is made. The endorsement contains a more advanced thumbprint about the bargained application that the browser confirms against an existing audit of the application's capabilities.
4) **ARP Spoofing:** Using this method, an attacker sends spoofed ARP messages over a network, which attempts to divert the traffic from its original host to the attacker's computer instead[2].
5) **Eavesdropping:** Eavesdropping can be active or passive and occurs through the interception of the traffic obtaining passwords, personal information and other sensitive information that are send over the traffic[3].
6) **Stealing Browser Cookies:** Cookies store information of login credentials, credit card information and other data that can be hijacked by the attacker using the MITM attack.

The damages and loss from these attacks can be vary depending on the motive of the attacker and the ability to deal with the data to cause mischief. With increase of the digital technology and use of Wireless communication everywhere in our daily life the consequences from these types of attacks are very serious. For example, "49 busted in Europe for Man-in-the-Middle bank attacks" where attackers were arrested on suspicion for using man-in-the-middle attacks to sniff out and intercept payment requests from email totaling fraud of €6 million[4]. It was this type of attack that hit the customers of one of South Africa's four large banks, Absa, in 2013.

II. Related work

Various works have been conducted in the field of network security each with its own particular techniques. In order to detect and prevent ARP spoofing attacks[5], they conducted a comprehensive survey, analyzed existing schemes, and identified the requirements for optimal solutions, this can be used as a reference by network administrators when deciding how to secure their LANs to ARP attacks.

It was [6] who presented new defence methods, and compared them with previously proposed approaches. Most schemes can be categorized as to their implementation methods (cryptographic, voting-based, hardware), and where they are located (server/host/client). An overview of some prominent defense mechanisms against DNS spoofing by off-path spoofing adversaries was conducted by ("Antidotes for DNS poisoning by off-path adversaries") and their taxonomy of those mechanisms was described as well.

Jorge Belenguer[7] presented a low-cost embedded IDS which, when plugged into a switch or hub, is able to detect and/or prevent MitM attacks automatically and efficiently. In this paper[8] A novel application of the ID based signature scheme has been proposed to prevent IP spoofing and Man-in-the-Middle Attack across a network by implementing with key generation, packet generation with signature, and signature verification modules. There are also other MITM prevention mechanisms like strong mutual authentication at endpoints, exchanging public keys in a secure channel, using secure certificates and advanced encryption in cryptography. There are even dedicated Wi-Fi sensors in the organization networks to detect the anomalies but no standard mechanism was implemented in the client side. This specified shortcoming in the MITM prevention mechanism is focused in this paper.

Mehmood et al., 2016 used different algorithms on a particular KDD99 dataset, which is the benchmark dataset used for anomaly-based detection technique and compared the performance measures with true positive rate, false positive rate, and precision[9]. Limthong et al., 2012 conducted experiments by studying the relationship between interval-based features of network traffic and several types of network anomalies by using two famous machine learning algorithms: the naıve Bayes and k-nearest neighbor[10].

Pu et al., 2020 presented a new hybrid unsupervised clustering-based anomaly detection method which gained popularity in machine learning-based cyber intrusion detection methods where this method combines Sub-Space Clustering (SSC) and One Class Support Vector Machine (OCSVM) to detect attacks without any prior knowledge and unknown types of attacks as well as zero-day attacks[11].

III. Methodology

An on-board wireless client (PC, Laptop) will generally search the nearby area for an SSID (SSID stands for Service Set Identifier, a name that clients and users can use to determine which access points are accessible). When a hacker stands nearby (like in the parking garage), he can use access points with high power (gain) antennas with the same SSID as the corporate network SSID, and respond to the client probe request with a valid response. The clients, as they typically associate with an access point with the highest power (signal strength), may get associated with the hacker's access point[12].

A wireless client, such as a laptop, mobile or tablet, will sense the air medium and scan for nearby access points with certain SSIDs (SSID represents the name of your wireless network, that is visible to anyone with a wireless device within reachable distance of your network.). An attacker can use such access points with high power antennas using the same SSID as the corporate network SSID and respond to client probe requests with a valid probe response. Due to the fact that the wireless client wants to associate with access points with the most power (signal strength), it is likely to associate with the attacker's access point.

As the access points send out a lot of data called beacon frames, they report their capability which usually includes SSID, channel, encryption type, authentication type, supported rates and a lot more as shown below. Based on preconfigured information or manually connecting to the network, once the wireless clients receive those Beacon frames, they determine whether the network they should connect to is the intended network. Once the wireless client has sent its probe request to the access point, and the access point has received the probe request, it will respond with a probe response containing data elements that access points can support. The connection will establish between the parties after the authentication and association phase. During those times, however, there is no identity information in the packets to validate their authenticity, making it possible for the attacker to create arbitrary packets impersonating any client or access point in the network.

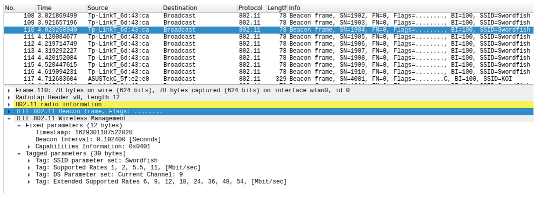
Wireless client decides which access point to connect to based on the below.

- **Pre-Configured network**: All the network connectivity information is already configured with keys or manually reconnect every time.
- Manually connecting to new network: Connection can be established by entering the network credentials
- **Preferred Network List**: A list of previously associated hotspots or access points where the client prioritize connecting to.

As discussed above, MITM attack happens when the attacker sends out the arbitrary beacon frames impersonating an access point that are in the network preferred list of the client, and after the client receive those frames, it will send the probe request and thereafter authentication and association only if the client sensing the air medium continuously. Continually scanning the network will enable the attacker to send beacon frames that the client without technical knowledge cannot identify as a threat. One weak link in the preferred network list will leave a window for hacker to create fake SSID which may end up connecting to client.

In order to prevent such type of attacks enterprises deploy heavy duty Wi-Fi sensors covering the perimeter of the enterprise network and continuously monitor the air and sends out to the central appliance where the machine learning algorithm detects the anomalies in the network and prevent attacks by de auth and several other techniques.

A very interesting discovery is the fact that airbase-ng and MDK3 use a hardcoded template for beacon frames generation that hasn't been changed since years/citeb13. Using simple rules, it is used to quickly recognize the attacker by going through the beacon frame information elements. We are taking into consideration the attacker beacon frame because the attacker does not know what the access point beacon frame looks like, and the attacker has cloning capabilities both in hardware and software. Since the parameters from the attacker beacon frames may differ from those from the legitimate access points, we will build an intrusion detection system using machine learning anomaly detection at the client side.



```
No.     Time         Source              Destination     Protocol  Length Info
    108 3.821869499  Tp-LinkT_6d:43:ca   Broadcast       802.11      78 Beacon frame, SN=1902, FN=0, Flags=........, BI=100, SSID=Swordfish
    109 3.921657196  Tp-LinkT_6d:43:ca   Broadcast       802.11      78 Beacon frame, SN=1903, FN=0, Flags=........, BI=100, SSID=Swordfish
    110 4.020260940  Tp-LinkT_6d:43:ca   Broadcast       802.11      78 Beacon frame, SN=1904, FN=0, Flags=........, BI=100, SSID=Swordfish
    111 4.120084677  Tp-LinkT_6d:43:ca   Broadcast       802.11      78 Beacon frame, SN=1905, FN=0, Flags=........, BI=100, SSID=Swordfish
    112 4.219714749  Tp-LinkT_6d:43:ca   Broadcast       802.11      78 Beacon frame, SN=1906, FN=0, Flags=........, BI=100, SSID=Swordfish
    113 4.319292227  Tp-LinkT_6d:43:ca   Broadcast       802.11      78 Beacon frame, SN=1907, FN=0, Flags=........, BI=100, SSID=Swordfish
    114 4.420152084  Tp-LinkT_6d:43:ca   Broadcast       802.11      78 Beacon frame, SN=1908, FN=0, Flags=........, BI=100, SSID=Swordfish
    115 4.520447615  Tp-LinkT_6d:43:ca   Broadcast       802.11      78 Beacon frame, SN=1909, FN=0, Flags=........, BI=100, SSID=Swordfish
    116 4.619094231  Tp-LinkT_6d:43:ca   Broadcast       802.11      78 Beacon frame, SN=1910, FN=0, Flags=........, BI=100, SSID=Swordfish
    117 4.712683604  ASUSTekC_5f:e2:e0   Broadcast       802.11     329 Beacon frame, SN=4081, FN=0, Flags=.......C, BI=100, SSID=KOI
▸ Frame 110: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface wlan0, id 0
▸ Radiotap Header v0, Length 12
▸ 802.11 radio information
▸ IEEE 802.11 Beacon frame, Flags: ........
▾ IEEE 802.11 Wireless Management
  ▾ Fixed parameters (12 bytes)
      Timestamp: 1629301187522020
      Beacon Interval: 0.102400 [Seconds]
    ▸ Capabilities Information: 0x0401
  ▾ Tagged parameters (30 bytes)
    ▸ Tag: SSID parameter set: Swordfish
    ▸ Tag: Supported Rates 1, 2, 5.5, 11, [Mbit/sec]
    ▸ Tag: DS Parameter set: Current Channel: 9
    ▸ Tag: Extended Supported Rates 6, 9, 12, 18, 24, 36, 48, 54, [Mbit/sec]
```

Fig. 2. A Typical Airbase-ng tool Beacon Frame.

The focus of this paper is on getting Wi-Fi telemetry data from a closed source OS instead of open-source OS because if we are able to get enough network information from a closed source OS, then it will be a piece of cake in an open-source software. With the default Windows Configurator, we do not get enough Wi Fi telemetry data, except for the SSIDs of access points. However, there is a built-in feature in Windows called Windows Event Logs, which may provide more information, but not enough to continue. In some cases, the command line can provide network information such as the mac address of the access point and the network profile, but not enough.

A. *Windows native API*

Windows has another feature called Windows Native API written in C++ which provides significant information about the Wi-Fi subsystem in Windows, such as state machine data, periodic scan results with BSS data, XML network profile data and card control like scan, connect, disconnect etc.. Native WiFi offers some native api functions that enable the use of functionality provided by the network cache files. Below is a list of these functions[14].

- **WlanGetNetworkBssList function** : The WlanGetNetworkBssList function retrieves a list of the basic service set (BSS) entries of the wireless network or networks on a given wireless LAN interface.
- **WlanGetProfileCustomUserData function**: The WlanGetProfileCustomUserData function gets the custom user data associated with a wireless profile.
- **WlanGetAvailableNetworkList function** : The WlanGetAvailableNetworkList function retrieves the list of available networks on a wireless LAN interface.

From the beginning of WLAN_BSS_ENTRY, this value indicates the offset in bytes for the information element (IE) data blob. It points to an information element (IE) buffer that contains variable-length information from 802.11 beacons or probe response frames. The IEs are from the last beacon or probe response frame received by the 802.11 BSS network. As soon as an IE appears in only one frame, the wireless LAN interface driver merges it with the other IEs from the most recent Beacon Response or Probe Response frame. There are three fields defining the general format of information elements in IEEE 802.11.1: an ID field containing 1 byte, a length field containing 1 byte, and a variable-length information field defining differences between elements. The

IEEE 802.11 standards assign a unique Element ID value to every information element. This binary blob data will give us information about the metadata of the beacon frames we see in the air.

As part of the dataset collection work, we used Chellam[15], which is an open source software that reconstructed everything contained in the beacon frame using API data from multiple sources. With those data elements we have information elements of each beacon frame in the vicinity. All of this information that the window subsystem is producing is stored and used in all types of detection, and they are generated entirely by the client, without any server appliance.

B. *Understanding the Attack Detection*

When we have enough information of every beacon frame in the surroundings, we can fingerprint the network with that data. Once this is accomplished, we can ensure that no one can spoof the access point, i.e. in your preferred network list. Metrics from the header fields of the beacon frame for fingerprinting is:

- BSSID(s)
- BSS type
- PHY type
- Beacon Interval
- Channels and Hopping
- Capability information
- Information Elements(s)
- Neighbouring Access Points

Once we have collected enough information about the surrounding beacon frames, we can fingerprint the network using this data. This ensures that someone cannot spoof the access point, for example in your preferred network list. Sometimes, a simple rule-based approach works, but it can be deceptively difficult to understand those rules for a person without much knowledge of how Wi-Fi works.

C. *Anomaly Detection using Machine Learning*

There has been a huge increase in use of machine learning algorithms for the detection of anomalies. These algorithms build a detection model or a prediction model using training data as input into the algorithm during the learning phase. The prediction model is then tested on a new set of data to differentiate between benign data and attack data. Input data need to be preprocessed before machine learning algorithms can understand them. Our input data contain examples, which we refer to as instances, observations or records[9]. Each record contains a set of attributes and features, which we call feature vectors. In this research, we used the above-mentioned metrics as features in our model.

D. *One-class SVM*

SVM is an algorithm originally developed for binary classification that can be applied to one-class classificationTo avoid bias when using for imbalanced classification, it is best to perform the standard SVM operation first, followed by the weighted SVM procedure. When modeling one class of data, the algorithm captures the density of the majority class and identifies outliers as those that fall under the extremes of the density function. This modification of SVM is referred to as One-Class SVM.

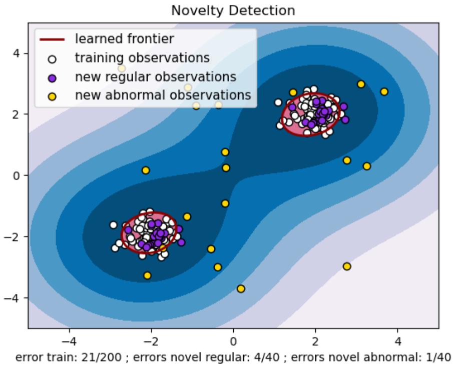


Fig. 3. One class Classification Model.

As of now, the data is trained just on one ”normal class,” such as authorized access points or attack tool families. As soon as the data is fed into the algorithm, the algorithm learns properties from the ”normal class.”. New Data points are tested against the trained data whenever they arrive and are given a +1 or -1 depending on whether they belong to the class or not. If they belong to the class, +1 is indicated and otherwise, -1 is indicated. The flow chart for the model is shown in Fig 4.

In this case, the metrics that uniquely define the beacon frame are extracted from the raw data gathered from the native API or network packet, and then sent to preprocessing where the data is sanitized, features are extracted, and then the data is converted to csv. With the sanitized data, we move onto the next phase called Feature Extraction and Transformation, where the features are selected, some are extracted and transformed for the algorithm. Data collected as part of this process is then fed into the One-class SVM model for training and testing in order to detect anomalies in the network.

IV. Results

To conduct this experiment, we used datasets from the cache memory of Windows laptops to simulate real-time operations. These data sets were cleaned by removing unnecessary fields and converted into csv for evaluation. We have selected metrics from the client system to represent the features of our model for fingerprinting the network since this research aims to demonstrate how MITM attacks can be prevented by using Machine Learning algorithms. Although our dataset only consists of 200 rows, we had many challenges while collecting the data, and it will take some more time to structure everything. The data is sent to the model for training and predictions. As can be seen from the Fig 5. the accuracy is 27.5%. Although the threshold may not be high enough for an efficient type of model, this is a

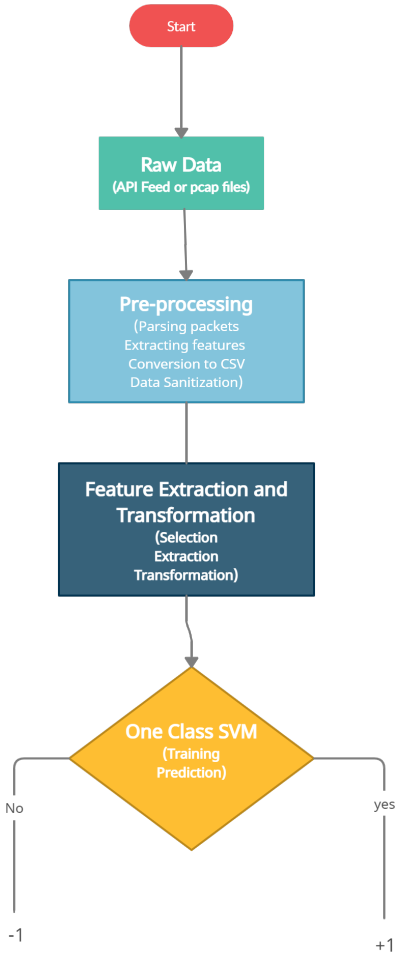
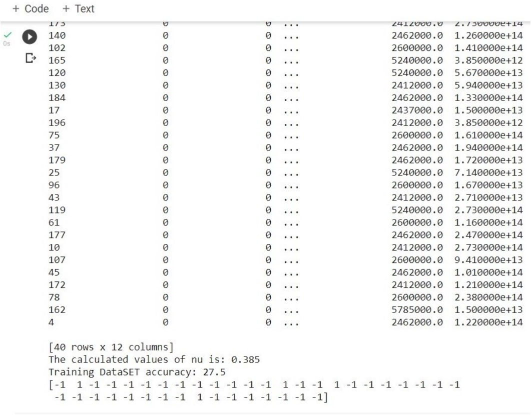


Fig. 4. Anomaly Detection Procedure.



```
173              0           0 ...          2412000.0  2.730000e+14
140              0           0 ...          2462000.0  1.260000e+14
102              0           0 ...          2600000.0  1.410000e+14
165              0           0 ...          5240000.0  3.850000e+12
120              0           0 ...          5240000.0  5.670000e+13
130              0           0 ...          2412000.0  5.940000e+13
184              0           0 ...          2462000.0  1.330000e+14
17               0           0 ...          2437000.0  1.500000e+13
196              0           0 ...          2412000.0  3.850000e+12
75               0           0 ...          2600000.0  1.610000e+14
37               0           0 ...          2462000.0  1.940000e+14
179              0           0 ...          2462000.0  1.720000e+13
25               0           0 ...          5240000.0  7.140000e+13
96               0           0 ...          2600000.0  1.670000e+13
43               0           0 ...          2412000.0  2.710000e+13
119              0           0 ...          5240000.0  2.730000e+14
61               0           0 ...          2600000.0  1.160000e+14
177              0           0 ...          2462000.0  2.470000e+14
10               0           0 ...          2412000.0  2.730000e+14
107              0           0 ...          2600000.0  9.410000e+13
45               0           0 ...          2462000.0  1.010000e+14
172              0           0 ...          2412000.0  1.210000e+14
78               0           0 ...          2600000.0  2.380000e+14
162              0           0 ...          5785000.0  1.500000e+13
4                0           0 ...          2462000.0  1.220000e+14

[40 rows x 12 columns]
The calculated values of nu is: 0.385
Training DataSET accuracy: 27.5
[-1  1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1  1 -1 -1  1 -1 -1 -1 -1 -1 -1 -1
 -1 -1 -1 -1 -1 -1 -1 -1  1 -1 -1 -1 -1 -1 -1 -1]
```

Fig. 5. Model Accuracy.

basic assumption that still requires a lot of improvement in the dataset and algorithms.

V. Discussion

While researching, several things came to my attention that helped me cope-up with mistakes and also learn new things in the process. In addition, I observed that some things contributed to the better performance of the model along with the parameters. Some of the observations include, but are not limited to:

- It is very hard to get enough telemetry data from a proprietary source software like Windows.
- There is high chance of the model being induced to detect the false positives, that has to be avoided.
- Even having native API, it is hard to embed all those scrambled data into a structure.
- I have yet to complete this research, but am trying to improve the precision and accuracy scores of the model as well as make it more efficient.

VI. CONCLUSION

A MITM attack can mimic a variety of security threats. We described different MITM defense mechanisms in this paper along with their descriptions and suggested a new intrusion detection approach using machine learning algorithms at client side. In summary, there are many plug and play methods to build an intrusion-based detection based on the parameters but we are mainly focusing on the beacon frames that are collected at client side autonomously and use those data to the machine learning algorithms to detect the anomalies in the network and prevent it from happening.

VII. FUTURE WORK

With machine learning algorithms, the model can be improved and taken into account. There are still a few more areas to test in the dataset. Plug and play with the features, Adding Interpretability, Neural Networks and Deep Learning aim at learning feature representations or anomaly scores for the sake of anomaly detection. An automated script can also be worked to collect data from the API's and convert them into csv with a definite structure.

VIII. ACKNOWLEDGMENT

With machine learning algorithms, the model can be improved and taken into account. There are still a few more areas to test in the dataset. Plug and play with the features, Adding Interpretability, Neural Networks and Deep Learning aim at learning feature representations or anomaly scores for the sake of anomaly detection. An automated script can also be worked to collect data from the API's and convert them into csv with a definite structure.