Cyber Attacks Classification using Machine Learning

Abstraction:

Cyber attacks are becoming increasingly sophisticated and difficult to detect using traditional security measures. Machine learning offers a promising approach to detecting and preventing cyber attacks by using advanced algorithms to identify patterns and anomalies in network traffic, system logs, and other data sources.

One abstraction for using machine learning to detect cyber attacks is to train a model on a large dataset of labeled examples of both normal and malicious network activity. The model can then be used to classify new data as either normal or suspicious based on the patterns it has learned.

Ultimately, the effectiveness of machine learning for detecting cyber attacks will depend on the quality of the data used to train the models, as well as the ability to continually update and adapt the models to new and evolving threats

Introduction:

Cyber attacks using machine learning (ML) are a growing concern in today's digital landscape. ML is a powerful tool that can be used to develop intelligent systems that can learn and adapt to new situations, making it a double-edged sword. On one hand, ML can be used to enhance cybersecurity by improving threat detection and response capabilities. On the other hand, it can also be used by cybercriminals to develop sophisticated attack strategies and evade traditional security measures.

In a cyber attack using ML, attackers can use machine learning algorithms to analyze large volumes of data and identify vulnerabilities in an organization's network. Once a vulnerability is identified, the attacker can use ML to develop custom malware or phishing attacks that are tailored to exploit the specific weakness in the network. ML can also be used to develop advanced social engineering tactics that can deceive users into revealing sensitive information or granting access to restricted systems.

As ML continues to evolve and become more accessible, the threat of cyber attacks using ML is likely to increase. It is important for organizations to implement strong security measures and stay vigilant against emerging threats in order to protect their networks and data from cyber attacks using machine learning.

Problem Formulation:

The problem formulation for a cyber attack using machine learning typically involves the following components:

1. Data collection: Collecting relevant data from various sources such as network logs, system logs, and other security-related data.
2. Data preprocessing: Cleaning, normalizing, and transforming the collected data into a suitable format for machine learning algorithms.
3. Feature extraction: Extracting relevant features from the preprocessed data that can be used by machine learning algorithms to detect cyber attacks.
4. Model selection: Selecting an appropriate machine learning algorithm that can effectively detect cyber attacks.
5. Model training: Training the selected machine learning algorithm on the preprocessed data to learn the patterns and characteristics of cyber attacks.
6. Model evaluation: Evaluating the performance of the trained machine learning algorithm on a separate dataset to determine its accuracy, precision, recall, and F1-score.
7. Model deployment: Deploying the trained machine learning algorithm in a real-world environment to detect cyber attacks in real-time.

Overall, the main objective of a cyber attack detection system using machine learning is to accurately identify and classify cyber attacks and alert security teams to take appropriate actions to mitigate the risks.

Dataset Description:

A data set for cyber attacks using machine learning typically includes a collection of various types of cyber attack data, such as network traffic, system logs, and malware samples. The data set is usually labeled, meaning that each attack is tagged with a category, such as ransomware, DDoS attack, or phishing attempt.
The data set may also include features extracted from the attack data, such as network packet headers, source IP addresses, and file hashes, which can be used as input for machine learning models.
In addition, the data set may include metadata such as the time and date of the attack, the severity of the attack, and the target of the attack. This information can be used to train models to detect and classify attacks, as well as to understand patterns in cyber attacks over time.

But the given data set has TWO files.
One have following information like:
['duration', 'protocol_type', 'service', 'flag', 'src_bytes'', 'land','urgent', 'hot', 'num_failed_logins', 'logged_in', 'num_compromised', 'num_file_creations', 'num_shells', 'num_access_files', 'is_host_login', 'is_guest_login', 'count', 'srv_count', ‘'attack_type', 'occcurance'] etc.

2nd file have:
['attack_type', 'occurrence'] etc.

First of all we split the given data into two files one for testing and other for training .

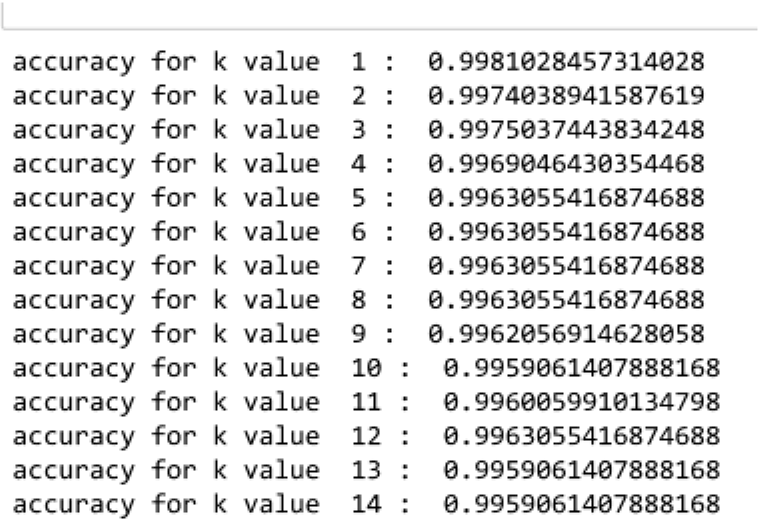
Then we map 2nd file data to training and testing files.Then we map (Attacks_types.txt) on 1st Training.txt and on 2nd Testingfile.txt
Then we find NaN values etc.
Then find outliers and values that can't use during computing like in KNN we did not use string but we used them as a result which is classification where data lie.

Classification and Clustering:

Classification is the process of dividing the data elements into specific classes based on their values. It is a type of supervised learning which means data are labeled. This project "Cyber-attack classification" uses commonly used machine learning classification algorithms to solve the problem by identifying the normal network traffics and attack classes.
We used there are 5 different classes so this comes under the classification problem where classification algorithms can be used to identify different classes based on attributes. In this report, I will discuss the various classification algorithms like KNN, DECISION TREE, MLP/ANN. We will be comparing their performance and time complexity.

k-nearest neighbors Classification (KNN):

k-nearest neighbors Classification (KNN) is a simple algorithm that classifies values based on similarity measures such as distance. It is used for both classification and regression predictive problems. However, it is mostly used in classification problems. This is simple and popular because it is easy to interpret the output. Similarly, calculation time is faster. It works by finding the distance between a point and data with the selected specific number closest to the point then votes for the most frequent label. Parameter n_neighbors is very important as it changes the accuracy rate. n_neighbors are a number of neighbors to use by default for neighbor's queries. There is no specific method to choose the best value for it. The default value of it is 5. One of the ways to find a better value for n_neighbors is to iterate n_neighbors values with some range (1 to 14) and check the accuracy. Once you know the accuracy of different values of n_neighbors, you can choose the higher accuracy value.

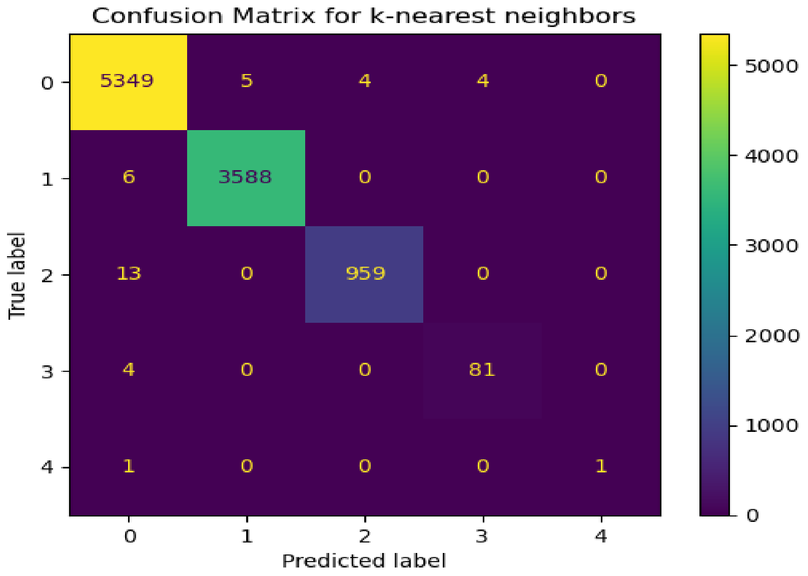


```
accuracy for k value  1 :  0.9981028457314028
accuracy for k value  2 :  0.9974038941587619
accuracy for k value  3 :  0.9975037443834248
accuracy for k value  4 :  0.9969046430354468
accuracy for k value  5 :  0.9963055416874688
accuracy for k value  6 :  0.9963055416874688
accuracy for k value  7 :  0.9963055416874688
accuracy for k value  8 :  0.9963055416874688
accuracy for k value  9 :  0.9962056914628058
accuracy for k value  10 :  0.9959061407888168
accuracy for k value  11 :  0.9960059910134798
accuracy for k value  12 :  0.9963055416874688
accuracy for k value  13 :  0.9959061407888168
accuracy for k value  14 :  0.9959061407888168
```

Now let me find out the better value for the n_neighbors parameter by running 1 to 14 and calculate the accuracy rate for each value. Here is the result when running value from 1 to 14. k (n_neighbors) value with 1 has a slightly higher accuracy rate than others so we will be using this value to train the model and see the difference.

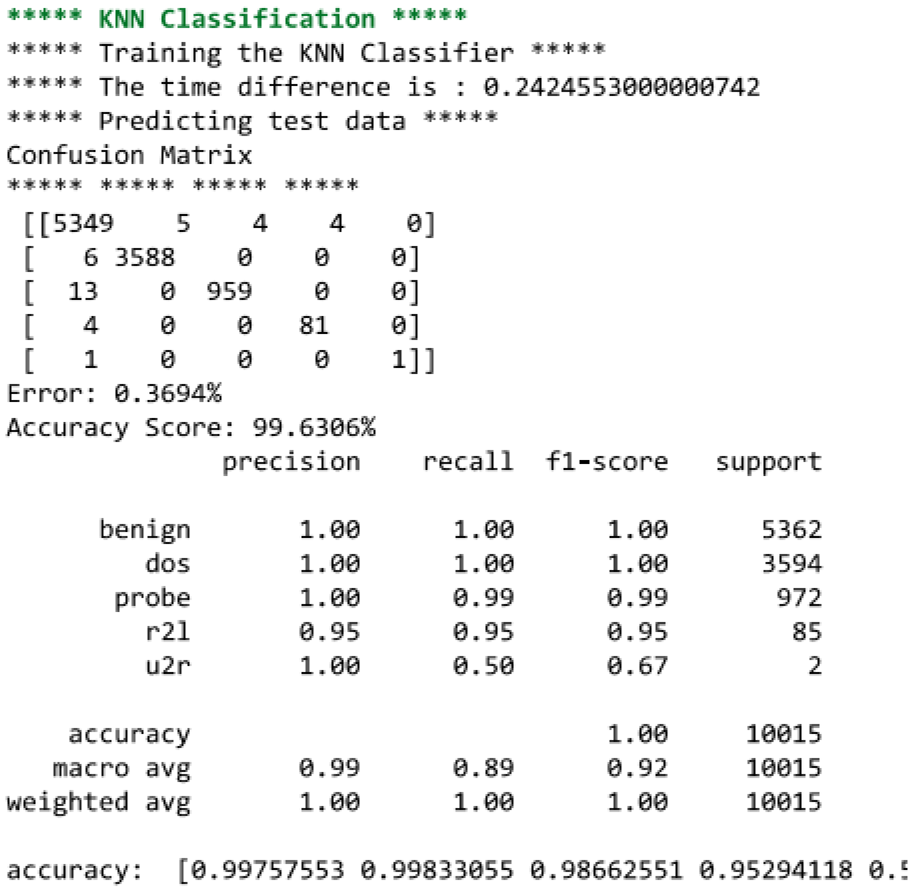
I have used an arbitrary value of 7 for n_neighbors and here is my confusion matrix. This confusion matrix of the k-nearest neighbors algorithm clearly shows that 5349 items are correctly identified as a benign class. Similarly, 3588 are correctly classified as dos, 959 as a probe, 81 as r2l and 1 as u2r. The performance measure of k-nearest neighbors Classifier algorithm is as follow

0 = Begin,   1 = Dos,   2 = Probe,   3 = r21,   4 = u2r



Confusion Matrix for k-nearest neighbors

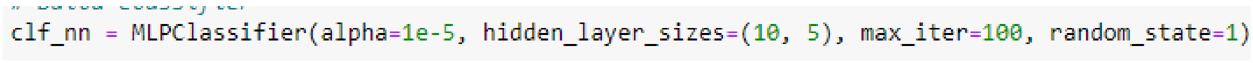
True label \ Predicted label	0	1	2	3	4
0	5349	5	4	4	0
1	6	3588	0	0	0
2	13	0	959	0	0
3	4	0	0	81	0
4	1	0	0	0	1



```
***** KNN Classification *****
***** Training the KNN Classifier *****
***** The time difference is : 0.2424553000000742
***** Predicting test data *****
Confusion Matrix
***** ***** ***** *****
 [[5349    5    4    4    0]
 [   6 3588    0    0    0]
 [  13    0  959    0    0]
 [   4    0    0   81    0]
 [   1    0    0    0    1]]
Error: 0.3694%
Accuracy Score: 99.6306%
              precision    recall  f1-score   support

      benign       1.00      1.00      1.00      5362
         dos       1.00      1.00      1.00      3594
       probe       1.00      0.99      0.99       972
         r2l       0.95      0.95      0.95        85
         u2r       1.00      0.50      0.67         2

    accuracy                           1.00     10015
   macro avg       0.99      0.89      0.92     10015
weighted avg       1.00      1.00      1.00     10015

accuracy:  [0.99757553 0.99833055 0.98662551 0.95294118 0.5
```

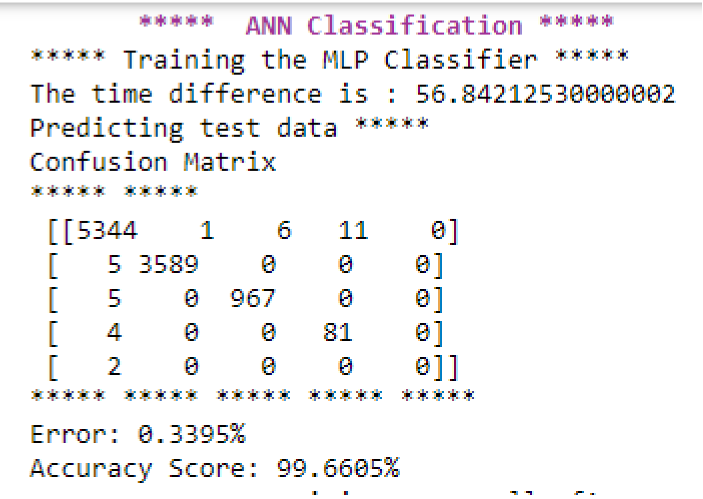
Neural Network:

Neural Network is a set of algorithms that can be used to recognise underlying relationships in a set of data through a process that mimics the way the human brain operates. In other words, a neural network is a system of neurons. It can adapt to changing input so the network generates the best possible result without needing to redesign the output criteria. Multi-layer Perceptrons (MLP) algorithm can be used to solve the classification problem where this algorithm trains using backpropagation. MLP classifier algorithm takes parameters such as hidden layers , activation, alpha and learning rate.

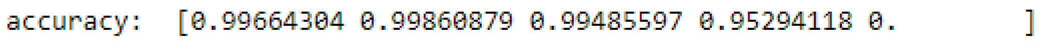


```
clf_nn = MLPClassifier(alpha=1e-5, hidden_layer_sizes=(10, 5), max_iter=100, random_state=1)
```

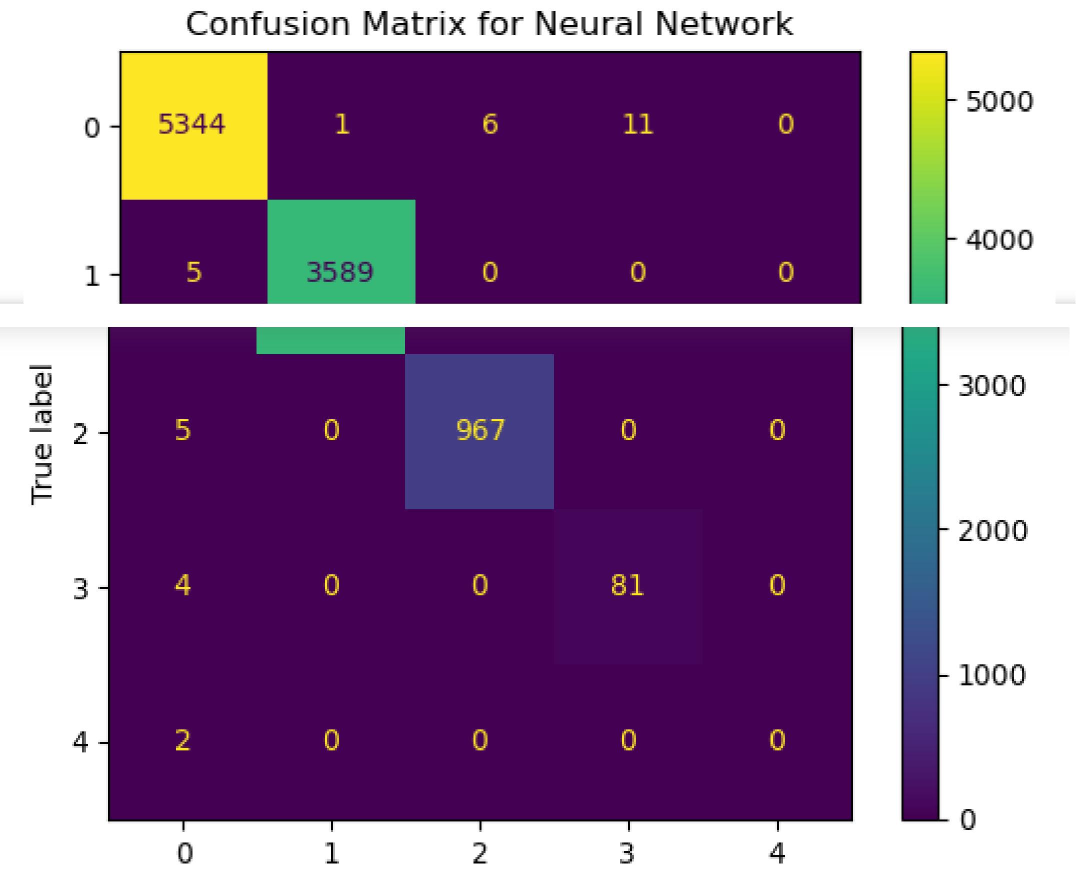
This confusion matrix of the neural network algorithm clearly shows that 5344 items are correctly identified as a benign class. Similarly, 3589 are correctly classified as dos, 967 as a probe,81 as r2l and did Not (0) identify any for u2r.



```
        *****  ANN Classification *****
***** Training the MLP Classifier *****
The time difference is : 56.84212530000002
Predicting test data *****
Confusion Matrix
***** *****
 [[5344    1    6   11    0]
 [   5 3589    0    0    0]
 [   5    0  967    0    0]
 [   4    0    0   81    0]
 [   2    0    0    0    0]]
***** ***** ***** ***** *****
Error: 0.3395%
Accuracy Score: 99.6605%
```

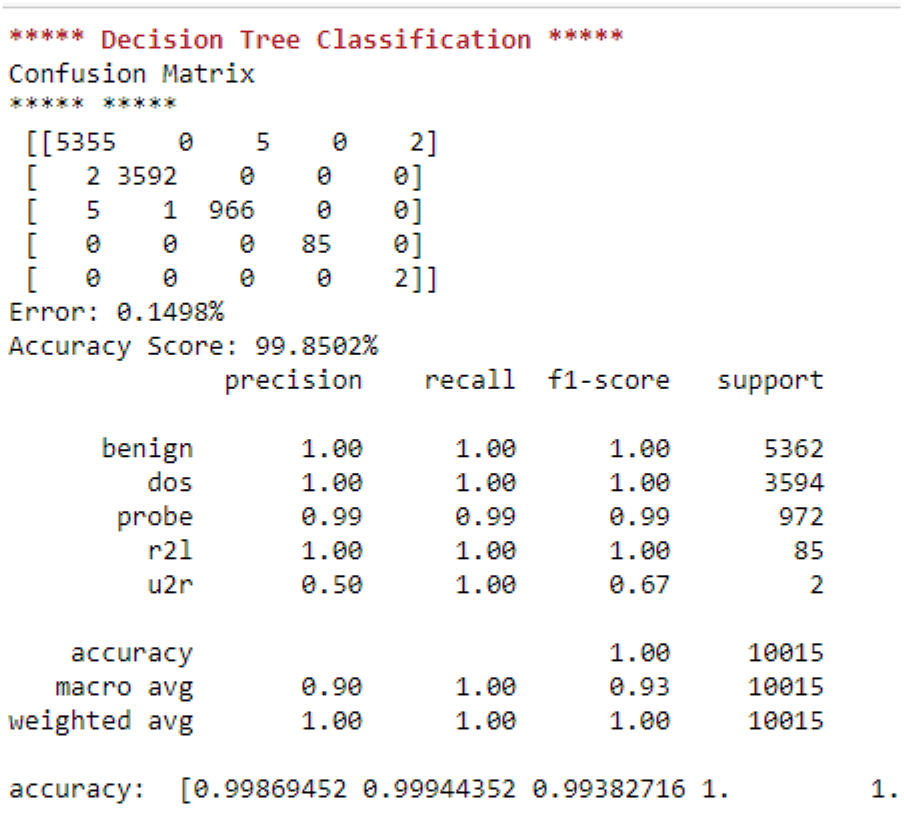


```
accuracy:  [0.99664304 0.99860879 0.99485597 0.95294118 0.        ]
```

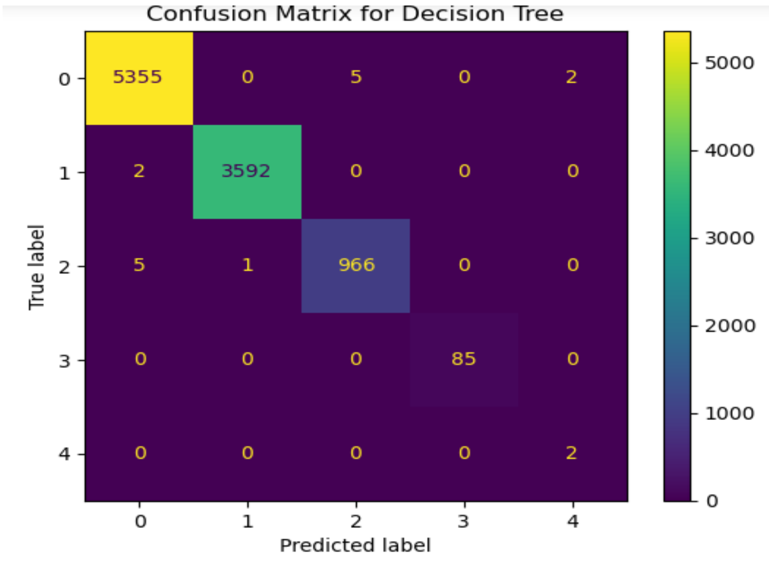


Decision Tree:

This confusion matrix of the neural network algorithm clearly shows that 5355 items are correctly identified as a benign class. Similarly, 3592 are correctly classified as dos, 966 as a probe,85 as r2l and 2 identify any for u2r.



```
***** Decision Tree Classification *****
Confusion Matrix
***** *****
 [[5355    0    5    0    2]
 [   2 3592    0    0    0]
 [   5    1  966    0    0]
 [   0    0    0   85    0]
 [   0    0    0    0    2]]
Error: 0.1498%
Accuracy Score: 99.8502%
              precision    recall  f1-score   support

      benign       1.00      1.00      1.00      5362
         dos       1.00      1.00      1.00      3594
       probe       0.99      0.99      0.99       972
         r2l       1.00      1.00      1.00        85
         u2r       0.50      1.00      0.67         2

    accuracy                           1.00     10015
   macro avg       0.90      1.00      0.93     10015
weighted avg       1.00      1.00      1.00     10015

accuracy:  [0.99869452 0.99944352 0.99382716 1.         1.
```

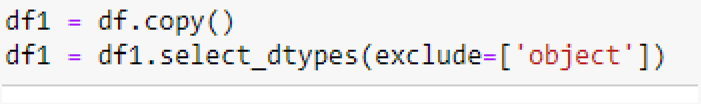


K-MEANS:

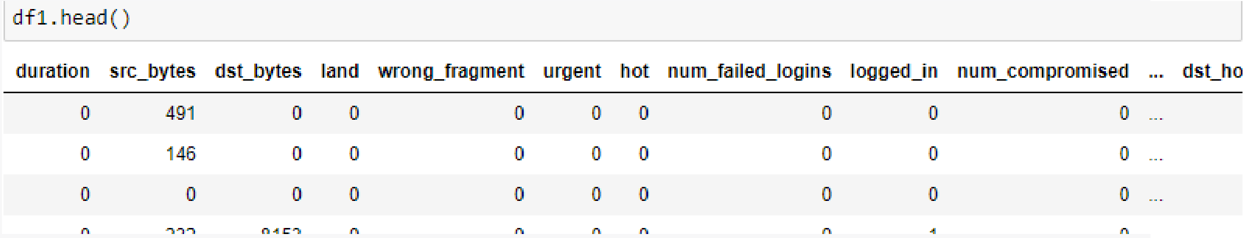
K-means clustering is a popular unsupervised machine learning algorithm that can be applied to cyber attack detection. The algorithm is used to group similar data points together based on their distance from a centroid. In the context of cyber attack detection, K-means clustering can be used to identify patterns and anomalies in network traffic data.

The following are the steps involved in applying K-means clustering to cyber attack detection:

Data preprocessing: In the case of K-means clustering, the data must be preprocessed to remove any noise, fill in any missing values, and normalize the data. We removed labels and then assigned them again.



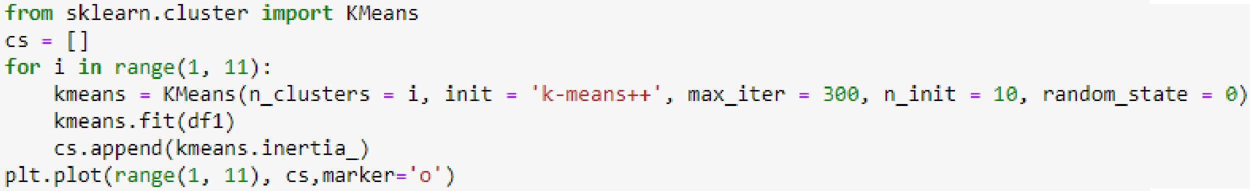
```
df1 = df.copy()
df1 = df1.select_dtypes(exclude=['object'])
```



```
df1.head()
```

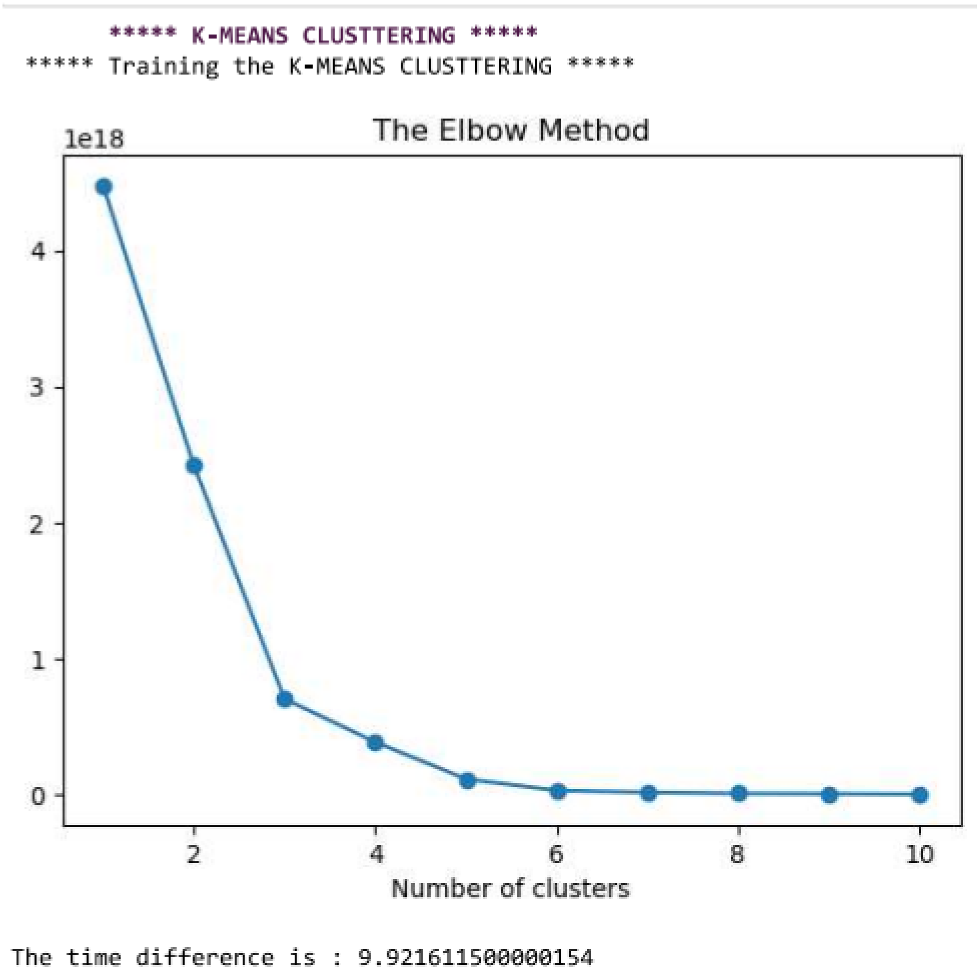
duration	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	num_failed_logins	logged_in	num_compromised	...	dst_ho
0	491	0	0	0	0	0	0	0	0	...	
0	146	0	0	0	0	0	0	0	0	...	
0	0	0	0	0	0	0	0	0	0	...	
0	232	8153	0	0	0	0	0	1	0		

Selecting the number of clusters: The K in K-means clustering represents the number of clusters that the algorithm should group the data points into. In the case of cyber attack detection, the number of clusters can be determined by the number of known attack types or by using other techniques such as the elbow method or silhouette analysis. We create 11 clusters.



```
from sklearn.cluster import KMeans
cs = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters = i, init = 'k-means++', max_iter = 300, n_init = 10, random_state = 0)
    kmeans.fit(df1)
    cs.append(kmeans.inertia_)
plt.plot(range(1, 11), cs,marker='o')
```

Initializing the centroids: The centroids are the points around which the algorithm groups the data. We iterated them 300 times and initially we assigned 10 clusters.Assigning data points to clusters



```
      ***** K-MEANS CLUSTTERING *****
 ***** Training the K-MEANS CLUSTTERING *****
```

```
The time difference is : 9.921611500000154
```

Conclusion:

Neural Network takes more time. In order to obtain great accuracy, outside of the scikit-learn can be explored such as Xgboost or Catboost which are based on booting algorithms. I also believe using ensemble learning algorithms can boost the performance of the models. Voting classification can also be used to combine multiple better performing algorithms and use a voting method which is the majority wins method for classification.