Attribution across Cyber Attack Types: Network Intrusions and Information Operations

Mr. V. S. Ramakrishna, Assoc Prof CSE:BVCE, ramakrishnavasamsetti@gmail.com
Mr. B. P. N. Madhukumar, Assoc Prof CSE:BVCE, bpnmadhukumar@gmail.com
Sangisetti Gowtham N C S Tarunkumar, CSE:BVCE
PullaRekhasri, CSE:BVCE
Rakesh Syam Sundar Guttula, CSE:BVCE
Mogalaturthi Kalyan Ram, CSE:BVCE

Article Info
Page Number: 5280 - 5299
Publication Issue:
Vol 71 No. 4 (2022)

Article History
***Article Received**: 25 March 2022*
***Revised**: 30 April 2022*
***Accepted**: 15 June 2022*
***Publication**: 19 August 2022*

Abstract

As the stakes get higher, intelligence and law enforcement agencies are working together to find the people responsible. This takes a lot of hard work. Tools and methods for figuring out who did bad things on the Internet are still in their early stages. Most of the time, people or groups are linked to attack activities through technical measurements, the origin of malicious code, and non-technical assessments of attack and attacker characteristics. Most of the time, figuring out who did an attack is a manual, time-consuming process that depends on both technical analysis and intelligence from the ground. As a result, this difficult and time-consuming process of attribution is mostly used for the worst cyber attacks and attacks on organisations with a lot of resources. Over time, we've gotten better at figuring out who did what. However, this is a double-edged sword: as attribution gets better, Internet privacy gets worse. This paper talks about attribution for two types of attacks that are at the centre of cyber conflict today: network intrusions and misinformation campaigns led by social bots. The paper talks about the current state of attribution for both types of attacks, makes suggestions for how it could be done better, and lays out directions for future research.

Cyber Attack, RFA, and Hyper parameter

Introduction

The Internet has become a big part of the social and economic life of the world. At the same time, cyber attacks are becoming more dangerous every year, making it a haven for crime and war. Anonymity is a key building block of the Internet. It comes from the need to hide users' identities and separate their actions from who they are. This makes it possible for people to say what they think without worrying about getting in trouble. This can help protect people from being watched by authoritarian governments. Privacy advocates have gone to great lengths to protect users' anonymity by giving them services like remailers and encryption, which make it harder for people to find out who they are. Online criminals and terrorists are also protected by this kind of anonymity. People who commit crimes like money laundering, extortion, espionage, and theft can hide behind a mask of anonymity. Actors in cyber warfare also use the anonymity of the Internet to spy, probe, and attack without drawing attention to themselves or being found out.

We talk about two types of cyber threats that are common right now: network intrusions and information operations that use social media (propagation of false narratives to manipulate opinion or behavior). We also talk about recent improvements in analysing and figuring out who was behind these incidents. We also make a plan for future research and ways to stop the cyber weapons race from getting worse through better attribution techniques.

1.1 Attribution is the key to legal proceedings because, without it, there are far fewer ways to hold people or countries accountable. Attributing cyber attacks is what gives a nation-state the right to fight back. But blaming cyber attacks on foreign governments in public is hard to do and can have serious legal and political consequences in some cases. Due to these complexities, it may not be necessary to know who did what in order to get even. Putting the blame for a cyber attack on a foreign government is often a calculated political move. Since the technical waters of attribution in cybercrime are always murky, seeking attribution is a political act, and any retaliation is less likely to be justified or seen in relation to the act itself. In fact, the U.S. government seems to be changing its mind about who should be blamed in public. In addition to blaming Iran and North Korea for attacks, the U.S. government has been more willing to blame Russia and the RIS for attacks (Russian Intelligence Services).

1.2 In a 2017 Joint Analytic Report (JAR), the Department of Homeland Security and the Federal Bureau of Investigation said, "Previous JARs have not linked malicious cyber activity to specific countries or threat actors. But technical signs from the U.S. Intelligence Community, DHS, FBI, the private sector, and other groups back up the idea that these activities are linked to RIS.

Legal rules for figuring out who did what don't take into account the complicated relationships between state and non-state actors or the fact that state actors often hide behind the actions of non-state actors to make it look like they didn't do it. Some people say that the only way to figure out who did what is to take a holistic and multidisciplinary approach.

1.2 Cyber attacks: Network intrusion and information operations are the two main types of cyber attacks.

Network intrusion is when someone breaks into a network illegally and in secret, usually for bad reasons like espionage, making things worse, or making money. There are different kinds of these.

- The first type of intrusion is used during nation-state wars, which are often accompanied by physical warfare. Its goal is to weaken the morale of the enemy country and overthrow its government by stopping citizens from communicating and spreading false information.
-

For information gathering and espionage, the second type of intrusion usually involves passive reconnaissance and probing of military, government, and private networks. This is done both to look for useful information and to find weaknesses in the networks and control systems, such as Supervisory Control and Data Acquisition (SCADA) systems.

The main reason for these kinds of activities is strategic. Getting this kind of information gives you technological know-how that can be used in business and the military in the future, lets you keep track of what a specific person is doing, and shows you where to attack if an enemy is hostile.

- The third kind of hacking tries to get into cyber-physical systems that depend more and more on information and communication systems. The goal of these kinds of attacks is to shut down cyber-physical systems like programmable logical controllers and SCADA systems, which make it possible for ATMs, power grids, and sewage treatment plants to work.

Information operations are actions taken by organised actors (governments or non-state actors) on social media and other Web platforms to change how people feel about politics at home or abroad, usually to achieve a strategic and/or geopolitical goal.

People all over the world use social media, and while it can be a good way to meet new people and share information, it also comes with some risks. For example, it makes it easy for almost anyone to share content that isn't likely to be checked for accuracy. This makes it a high-risk way for false information to spread.

People can use social bots to do bad things in a number of different ways. These bots are used to control social media services (SMS) by making content, sharing posts, and even commenting on posts made by real people.

1.3 Motivation: The goal of this study is to find out how machine learning algorithms can help better attribute cyber warfare and how combining these models can make them work better than they do now, since there isn't a single paper that talks about how cyber attacks are attributed in this way. Lastly, it's important to know and understand how these models can be used to make predictions that are different from one another.

1.4 The main problem and challenge is attribution in cyber warfare and spoofing of data. For example, a fake address of origin can be used in packets to hide the real IP-address, which makes the problem of identification even harder.

1.5 The goal of this paper is to explain the problem of figuring out who did surveillance, data theft, espionage, and misinformation campaigns as part of cyber warfare.

1.6 Scope: The goal of this work is to make it easier to know who did what on the internet and to protect people's privacy.

1.7 How to Use:

• Communication: In the world we live in now, we can send and receive any message through e-mail, and we don't have to use postage stamps.

• Social networking: These apps are very popular with teens and young adults. It could one day take the place of physical networking.

• Opportunities: Social media can be used for more than just making friends. We can get different job offers and choose the one we want.

3: Proposed System: In this paper, we use the Network Intrusion and Fake News datasets to train a machine learning algorithm called Random Forest. This trained model can be used to predict whether a network packet or social media news is a fake/attack or not.

3.1 Algorithm:

3.1.1 algorithm for the random forest:

Random forests, also called random decision forests, are a type of ensemble learning method that can be used for classification, regression, and other tasks. It works by building a lot of decision trees at training time. For classification tasks, the random forest gives the class chosen by the most trees as the answer.

This model is based on three random ideas: picking training data at random when making trees, picking some subsets of features when splitting nodes, and only looking at a subset of all features when splitting each node in each simple decision tree. In a random forest, each tree learns from a random sample of the data points when it is being trained.

- It is a classifier that has a number of decision trees on different subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset.
- The more trees there are in the forest, the better the accuracy and the less likely it is that the model will be too good.

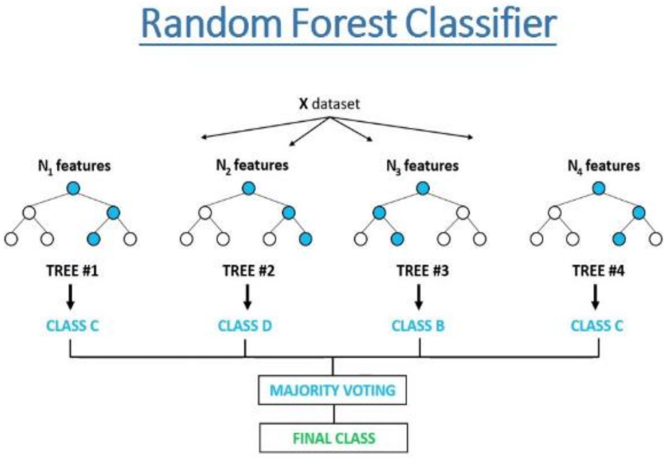


Fig.1. random forest classifier

Hyperparameters of Random Forest:

- Firstly, there is the **n estimators** hyperparameter, which is just the number of trees the algorithm builds before taking the maximum voting or taking the averages of predictions. In general, a higher number of trees increases the performance and makes the predictions more stable, but it also slows down the computation.
- Another important hyperparameter is **max features,** which is the maximum number of features random forest considers to split a node. Sklearn provides several options, all described in the documentation.

- The last important hyperparameter is **min_sample_leaf.** This determines the minimum number of leafs required to split an internal node.

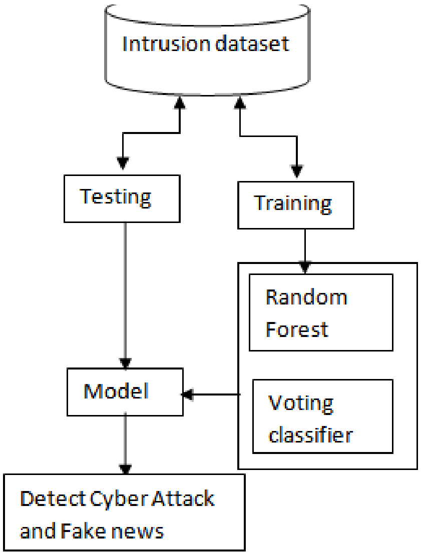


Fig.2

3.4 Algorithm and Process design:

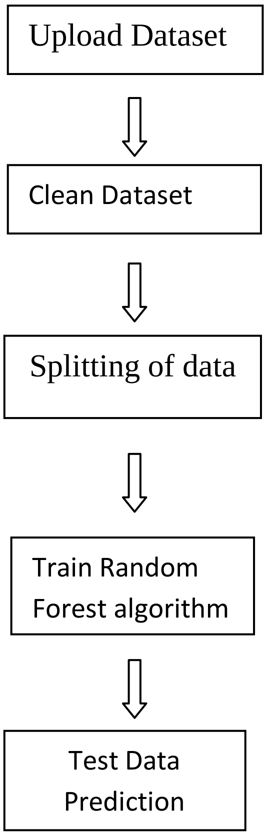


Fig.3. Process Design

Modules:

Load Dataset: We will load datasets into applications using this module.

Clean Dataset: Every dataset has missing and non-numeric data, and machine learning algorithms can only work with numeric data. To clean data, we need to get rid of missing values and change non-numeric data to numbers. Split dataset into train and test: split dataset into 80% training and 20% testing. Train machine learning algorithm on 80% data, then apply trained model to 20% data for prediction. Compare predicted values with original data to see how accurate algorithm prediction is.

Train the Random Forest model: In this module, we'll use the RF algorithm to feed train data into the model. Test data prediction: With this module, we will use test data on a trained model to predict if the test data is NORMAL or if it contains an attack or fake news.

4 Methods Used and Results: 4.1 Concerning the data:

In this project, we use a "intrusion dataset" and a "fake news dataset." The dataset is an important part of building IDS models based on machine learning. This dataset gives a wide range of simulated intrusions into a military network. By imitating a typical US Air Force LAN, it made a place where raw TCP/IP dump data for a network could be collected. The LAN was set up to look like a real place and attacked multiple times. A connection is a series of TCP packets that start and end at a certain time and allow data to flow from an IP address of origin to an IP address of destination using a well-defined protocol. Also, each connection is marked as either normal or an attack, and there is only one type of attack for each connection. About 100 bytes are used for each connection record.

Normal and attack data are used to get 41 quantitative and qualitative features for each TCP/IP connection. There are 3 qualitative features and 38 quantitative features.

There are two groups in the class variable:

• Normal • Anomalous

The first step is to get traffic from the internet as either a packet or a flow. After that, the traffic is put together into a certain type of data that includes labelling and other network-related features.

Dataset on Fake News:

There are two kinds of articles in the dataset: fake News and real News. This set of data was gathered from real-world sources. For example, the truthful articles were found by crawling Reuters.com (News website). As for the fake news stories, they came from a variety of places. The fake news stories came from websites that Politifact, a fact-checking group in the United States, and Wikipedia said were not reliable. The dataset has different kinds of articles about different topics, but most of the articles are about politics and World news. There are two CSV files that make up the dataset. More than 12,600 articles from reuters.com are in the first file, which is called "True.csv." The second file, called "Fake.csv," has more than 12,600 fake news articles from different sources. Each article has a title, text, type, and the date it was published. To match the fake news data that Kaggle.com collected. The data that was collected was cleaned up and put to use, but the fake news kept its punctuation and spelling mistakes.

Measures of Performance:

In this graph, the x-axis shows the type of protocol, and the y-axis shows how many packets of that protocol type there are. In this graph, the x-axis shows the true class and the y-axis shows the predicted class. This is a confusion matrix for predicting attacks and normal behaviour.

Confusion Matrix is a useful machine learning method that lets you measure Recall, Precision, Accuracy, and the AUC-ROC curve.

For this, values are calculated based on: • True positive (TP) = number of events for which the correct answer was given.

- False negative (FN): The number of events that were wrongly predicted and did not happen.
- False-positive (FP) = the number of wrongly predicted events.
- True negative (TN): The number of events that could have happened but didn't.

Precision is the number of correct positive predictions out of the total number of positive predictions.

Accuracy = TP/TP + FP

Remember that it is the number of positive observations that can be predicted correctly out of all the observations in the original data.

Recall = TP/TP plus FN

F1-score: It is the average of Precision and Recall with a weighted average. So, both false positives and false negatives are taken into account in this score. It is not as easy to understand as accuracy, but F Measure is usually more useful than accuracy, especially if you have a rough idea of how the class is distributed. The best way for accuracy to work is if both false positives and false negatives cost the same. If the prices for Precision and Recall are different, it's best to look at both.

F1 Score = 2(Precision Recall/Precision + Recall)

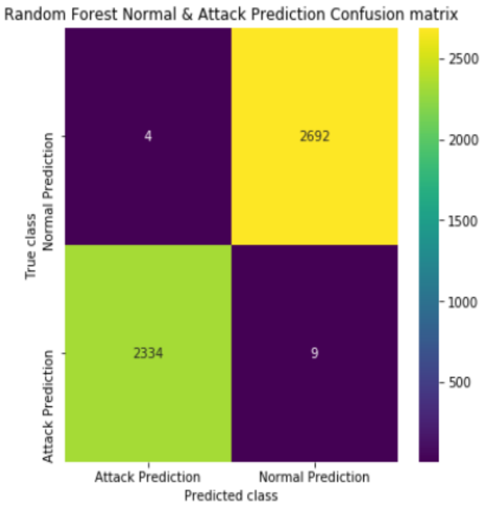
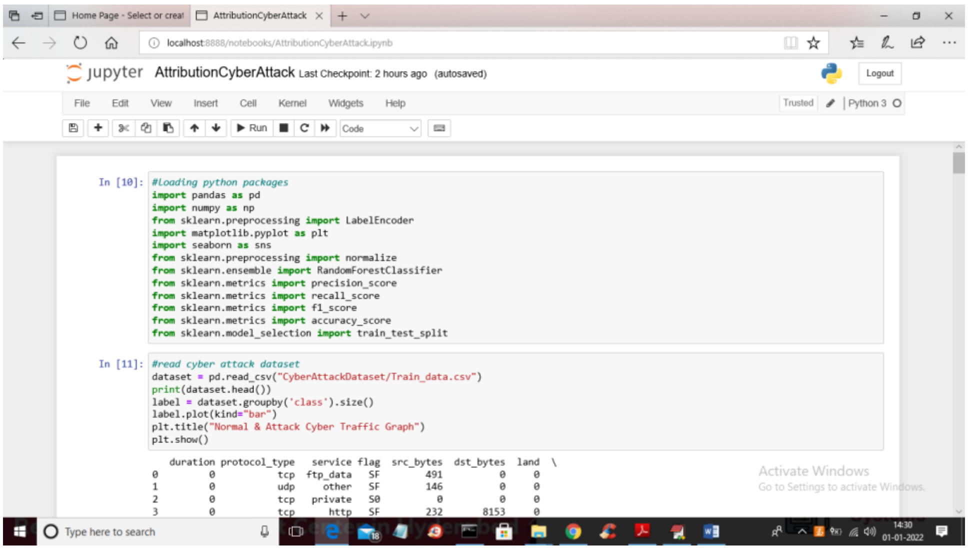


Fig.4

4.3 Outcome:

In propose paper we are training machine learning algorithm called Random Forest with Network Intrusion and FAKE NEWS dataset and this trained model can be used to predict network packet or social media news as fake/attack or genuine.

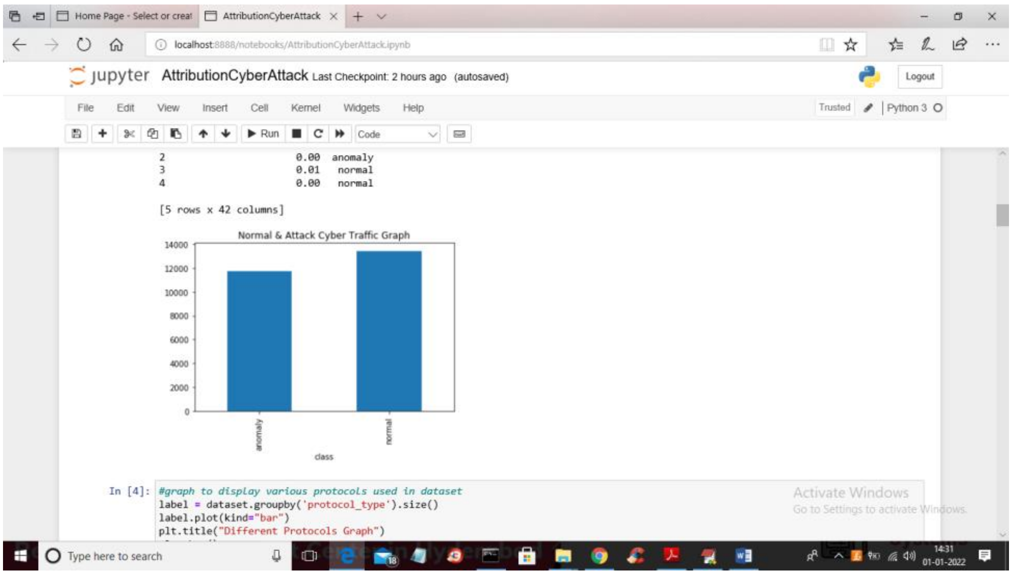
In below figure we are loading require python packages and then reading and displaying dataset values


Home Page - Select or creat
AttributionCyberAttack
localhost:8888/notebooks/AttributionCyberAttack.ipynb
jupyter
AttributionCyberAttack Last Checkpoint: 2 hours ago (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code
Activate Windows
Go to Settings to activate Windows.
Type here to search
14:30
01-01-2022

```
In [10]: #Loading python packages
         import pandas as pd
         import numpy as np
         from sklearn.preprocessing import LabelEncoder
         import matplotlib.pyplot as plt
         import seaborn as sns
         from sklearn.preprocessing import normalize
         from sklearn.ensemble import RandomForestClassifier
         from sklearn.metrics import precision_score
         from sklearn.metrics import recall_score
         from sklearn.metrics import f1_score
         from sklearn.metrics import accuracy_score
         from sklearn.model_selection import train_test_split

In [11]: #read cyber attack dataset
         dataset = pd.read_csv("CyberAttackDataset/Train_data.csv")
         print(dataset.head())
         label = dataset.groupby('class').size()
         label.plot(kind="bar")
         plt.title("Normal & Attack Cyber Traffic Graph")
         plt.show()

            duration protocol_type   service flag  src_bytes  dst_bytes  land  \
         0         0           tcp  ftp_data   SF        491          0     0
         1         0           udp     other   SF        146          0     0
         2         0           tcp   private   S0          0          0     0
         3         0           tcp      http   SF        232       8153     0
```

Fig.5

In below figure we are displaying total number of NORMAL and ATTACK packets found in dataset

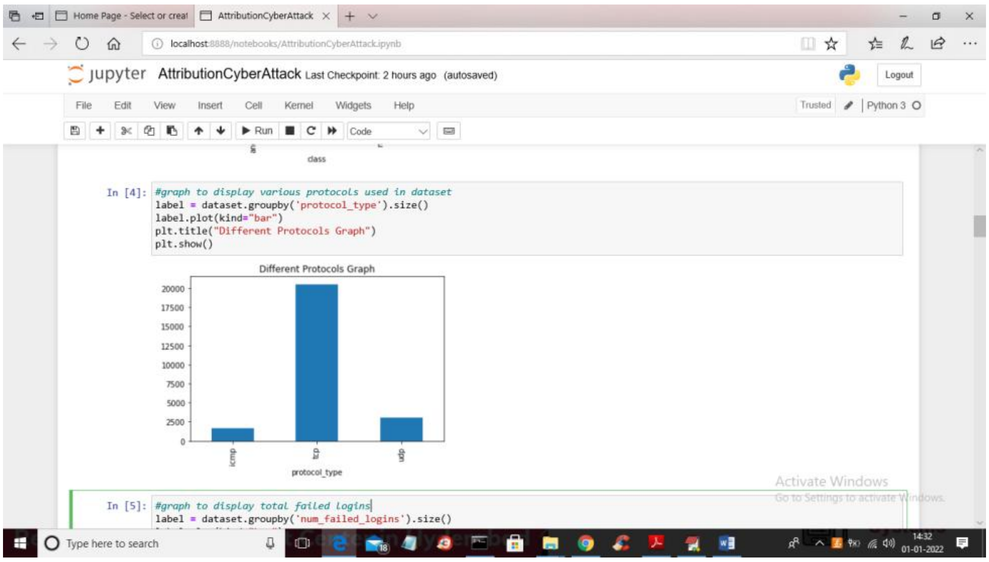


```
2                          0.00  anomaly
3                          0.01   normal
4                          0.00   normal

[5 rows x 42 columns]
```

```
In [4]: #graph to display various protocols used in dataset
        label = dataset.groupby('protocol_type').size()
        label.plot(kind="bar")
        plt.title("Different Protocols Graph")
```

Fig.6

In above graph X-axis represents type of packet and y-axis represents number of records of that type. In below figure I am displaying graph for different types of protocols found in dataset

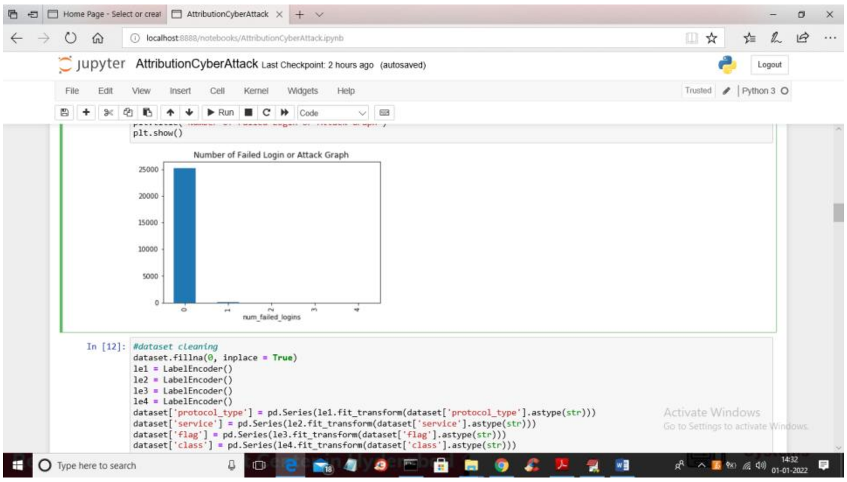


```
In [4]: #graph to display various protocols used in dataset
        label = dataset.groupby('protocol_type').size()
        label.plot(kind="bar")
        plt.title("Different Protocols Graph")
        plt.show()
```

```
In [5]: #graph to display total failed logins
        label = dataset.groupby('num_failed_logins').size()
```

Fig.7

In above graph x-axis represents protocol type and y-axis represents total packets of that protocol type. In below figure we are displaying graph for failed login

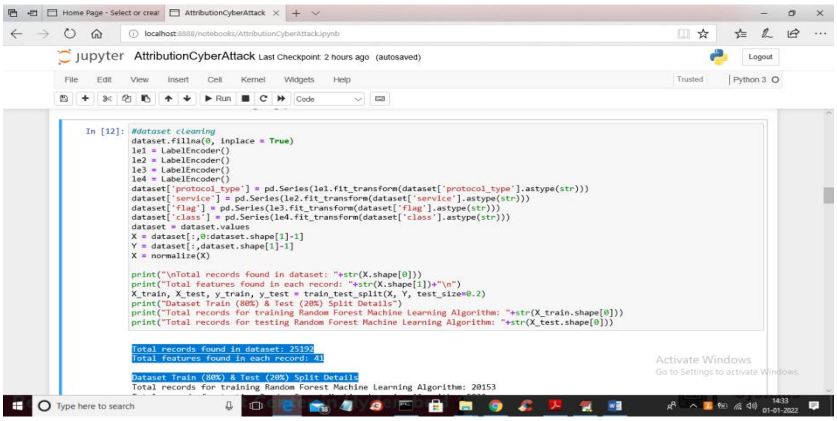

AttributionCyberAttack Last Checkpoint: 2 hours ago (autosaved)
Number of Failed Login or Attack Graph
25000
20000
15000
10000
5000
0
0
1
2
3
4
num_failed_logins

```
plt.show()
```

```
In [12]: #dataset cleaning
dataset.fillna(0, inplace = True)
le1 = LabelEncoder()
le2 = LabelEncoder()
le3 = LabelEncoder()
le4 = LabelEncoder()
dataset['protocol_type'] = pd.Series(le1.fit_transform(dataset['protocol_type'].astype(str)))
dataset['service'] = pd.Series(le2.fit_transform(dataset['service'].astype(str)))
dataset['flag'] = pd.Series(le3.fit_transform(dataset['flag'].astype(str)))
dataset['class'] = pd.Series(le4.fit_transform(dataset['class'].astype(str)))
```

Fig.8

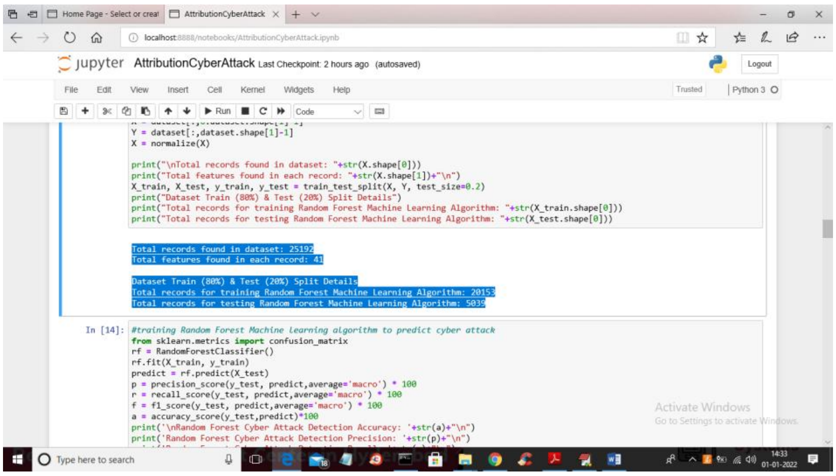
In below figure we are cleaning dataset and then splitting dataset into train and test


AttributionCyberAttack Last Checkpoint: 2 hours ago (autosaved)

```
In [12]: #dataset cleaning
dataset.fillna(0, inplace = True)
le1 = LabelEncoder()
le2 = LabelEncoder()
le3 = LabelEncoder()
le4 = LabelEncoder()
dataset['protocol_type'] = pd.Series(le1.fit_transform(dataset['protocol_type'].astype(str)))
dataset['service'] = pd.Series(le2.fit_transform(dataset['service'].astype(str)))
dataset['flag'] = pd.Series(le3.fit_transform(dataset['flag'].astype(str)))
dataset['class'] = pd.Series(le4.fit_transform(dataset['class'].astype(str)))
dataset = dataset.values
X = dataset[:,0:dataset.shape[1]-1]
Y = dataset[:,dataset.shape[1]-1]
X = normalize(X)

print("\nTotal records found in dataset: "+str(X.shape[0]))
print("Total features found in each record: "+str(X.shape[1])+"\n")
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2)
print("Dataset Train (80%) & Test (20%) Split Details")
print("Total records for training Random Forest Machine Learning Algorithm: "+str(X_train.shape[0]))
print("Total records for testing Random Forest Machine Learning Algorithm: "+str(X_test.shape[0]))
```

```
Total records found in dataset: 25192
Total features found in each record: 41

Dataset Train (80%) & Test (20%) Split Details
Total records for training Random Forest Machine Learning Algorithm: 20153
```

Fig.9


AttributionCyberAttack Last Checkpoint: 2 hours ago (autosaved)

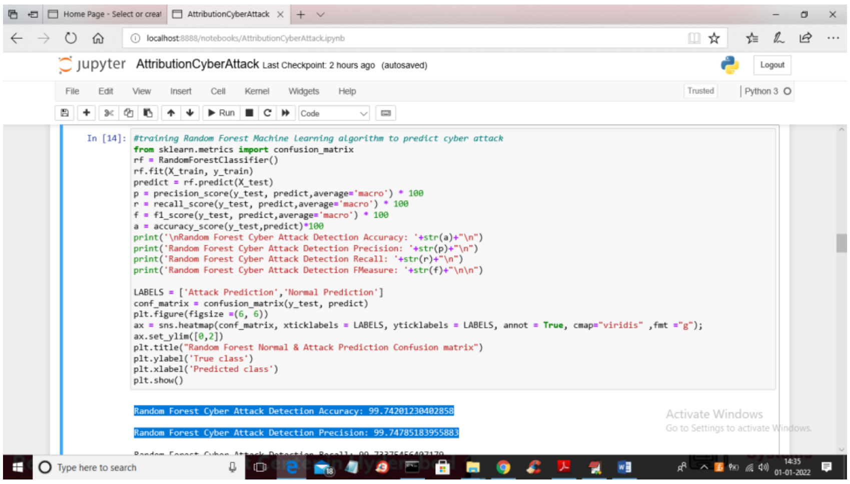
```
Y = dataset[:,dataset.shape[1]-1]
X = normalize(X)

print("\nTotal records found in dataset: "+str(X.shape[0]))
print("Total features found in each record: "+str(X.shape[1])+"\n")
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2)
print("Dataset Train (80%) & Test (20%) Split Details")
print("Total records for training Random Forest Machine Learning Algorithm: "+str(X_train.shape[0]))
print("Total records for testing Random Forest Machine Learning Algorithm: "+str(X_test.shape[0]))
```

```
Total records found in dataset: 25192
Total features found in each record: 41

Dataset Train (80%) & Test (20%) Split Details
Total records for training Random Forest Machine Learning Algorithm: 20153
Total records for testing Random Forest Machine Learning Algorithm: 5039
```

```
In [14]: #training Random Forest Machine Learning algorithm to predict cyber attack
from sklearn.metrics import confusion_matrix
rf = RandomForestClassifier()
rf.fit(X_train, y_train)
predict = rf.predict(X_test)
p = precision_score(y_test, predict,average='macro') * 100
r = recall_score(y_test, predict,average='macro') * 100
f = f1_score(y_test, predict,average='macro') * 100
a = accuracy_score(y_test,predict)*100
print('\nRandom Forest Cyber Attack Detection Accuracy: '+str(a)+"\n")
print('Random Forest Cyber Attack Detection Precision: '+str(p)+"\n")
```

Fig.10

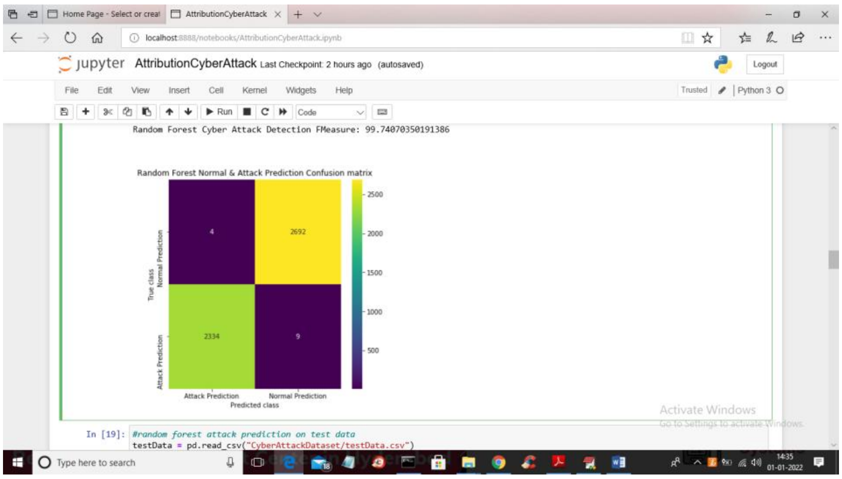
In above two figures we can see dataset size with train and test split and in below figure we are training Random Forest with training data


Home Page - Select or creat
AttributionCyberAttack
localhost:8888/notebooks/AttributionCyberAttack.ipynb
jupyter AttributionCyberAttack Last Checkpoint: 2 hours ago (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code
Activate Windows
Go to Settings to activate Windows.
Type here to search
14:35
01-01-2022

```
In [14]: #training Random Forest Machine Learning algorithm to predict cyber attack
         from sklearn.metrics import confusion_matrix
         rf = RandomForestClassifier()
         rf.fit(X_train, y_train)
         predict = rf.predict(X_test)
         p = precision_score(y_test, predict,average='macro') * 100
         r = recall_score(y_test, predict,average='macro') * 100
         f = f1_score(y_test, predict,average='macro') * 100
         a = accuracy_score(y_test,predict)*100
         print('\nRandom Forest Cyber Attack Detection Accuracy: '+str(a)+"\n")
         print('Random Forest Cyber Attack Detection Precision: '+str(p)+"\n")
         print('Random Forest Cyber Attack Detection Recall: '+str(r)+"\n")
         print('Random Forest Cyber Attack Detection FMeasure: '+str(f)+"\n\n")

         LABELS = ['Attack Prediction','Normal Prediction']
         conf_matrix = confusion_matrix(y_test, predict)
         plt.figure(figsize =(6, 6))
         ax = sns.heatmap(conf_matrix, xticklabels = LABELS, yticklabels = LABELS, annot = True, cmap="viridis" ,fmt ="g");
         ax.set_ylim([0,2])
         plt.title("Random Forest Normal & Attack Prediction Confusion matrix")
         plt.ylabel('True class')
         plt.xlabel('Predicted class')
         plt.show()

         Random Forest Cyber Attack Detection Accuracy: 99.74201230402858

         Random Forest Cyber Attack Detection Precision: 99.74785183955883

         Random Forest Cyber Attack Detection Recall: 99.73375456407179
```

Fig.11

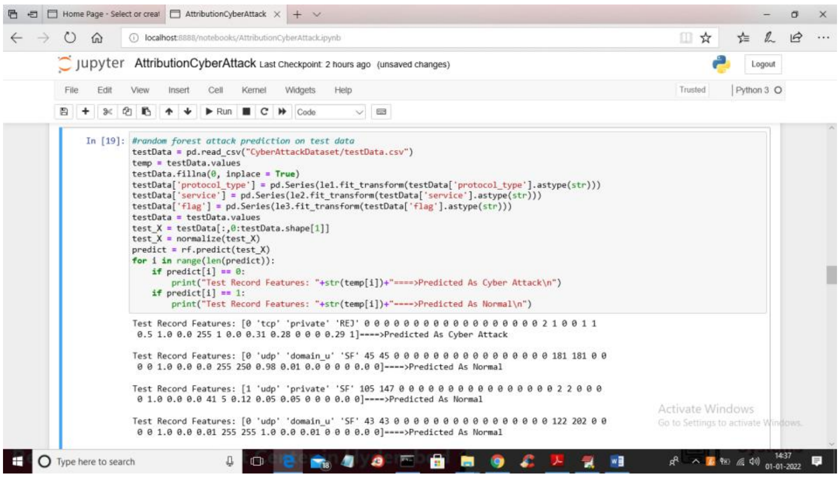
In above figure we are training data with random forest and we got accuracy of random forest as 99% and in below figure we can see confusion matrix of predicted and original data



```
Random Forest Cyber Attack Detection FMeasure: 99.74070350191386

In [19]: #random forest attack prediction on test data
         testData = pd.read_csv("CyberAttackDataset/testData.csv")
```

Fig.12

In above graph only 4 and 9 records are incorrectly predicted and in below figure we can see random forest prediction on new test data



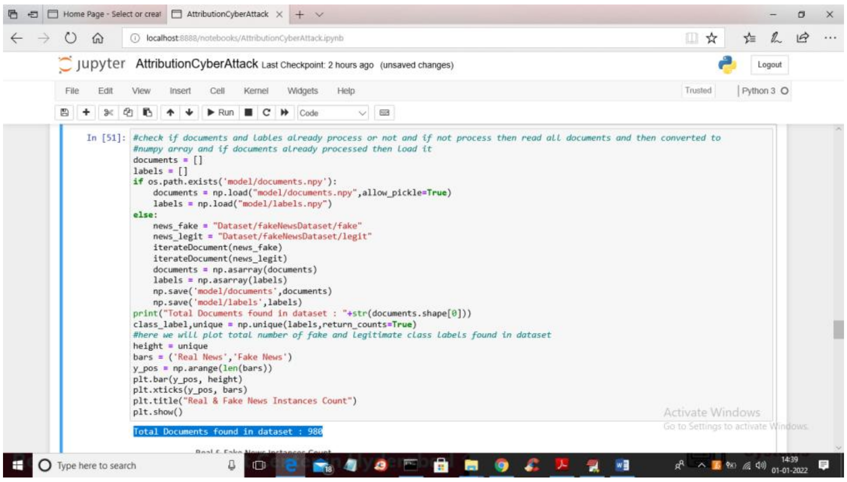
Home Page - Select or creat
AttributionCyberAttack
localhost:8888/notebooks/AttributionCyberAttack.ipynb
jupyter AttributionCyberAttack Last Checkpoint: 2 hours ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Activate Windows
Go to Settings to activate Windows.
Type here to search
14:37
01-01-2022

```
In [19]: #random forest attack prediction on test data
         testData = pd.read_csv("CyberAttackDataset/testData.csv")
         temp = testData.values
         testData.fillna(0, inplace = True)
         testData['protocol_type'] = pd.Series(le1.fit_transform(testData['protocol_type'].astype(str)))
         testData['service'] = pd.Series(le2.fit_transform(testData['service'].astype(str)))
         testData['flag'] = pd.Series(le3.fit_transform(testData['flag'].astype(str)))
         testData = testData.values
         test_X = testData[:,0:testData.shape[1]]
         test_X = normalize(test_X)
         predict = rf.predict(test_X)
         for i in range(len(predict)):
             if predict[i] == 0:
                 print("Test Record Features: "+str(temp[i])+"====>Predicted As Cyber Attack\n")
             if predict[i] == 1:
                 print("Test Record Features: "+str(temp[i])+"---->Predicted As Normal\n")

         Test Record Features: [0 'tcp' 'private' 'REJ' 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 1 0 0 1 1
          0.5 1.0 0.0 255 1 0.0 0.31 0.28 0 0 0 0.29 1]---->Predicted As Cyber Attack

         Test Record Features: [0 'udp' 'domain_u' 'SF' 45 45 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 181 181 0 0
          0 0 1.0 0.0 0.0 255 250 0.98 0.01 0.0 0 0 0 0.0 0]---->Predicted As Normal

         Test Record Features: [1 'udp' 'private' 'SF' 105 147 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 0 0 0
          0 1.0 0.0 0.0 41 5 0.12 0.05 0.05 0 0 0 0.0 0]---->Predicted As Normal

         Test Record Features: [0 'udp' 'domain_u' 'SF' 43 43 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 122 202 0 0
          0 0 1.0 0.0 0.01 255 255 1.0 0.0 0.01 0 0 0 0.0 0]---->Predicted As Normal
```

Fig.13

In above figure in output before =➔ arrow symbol I am displaying test data and after arrow symbol I am displaying predicted results as CYBER ATTACK or NORMAL.

Now we are training random forest with FAKE NEWS dataset

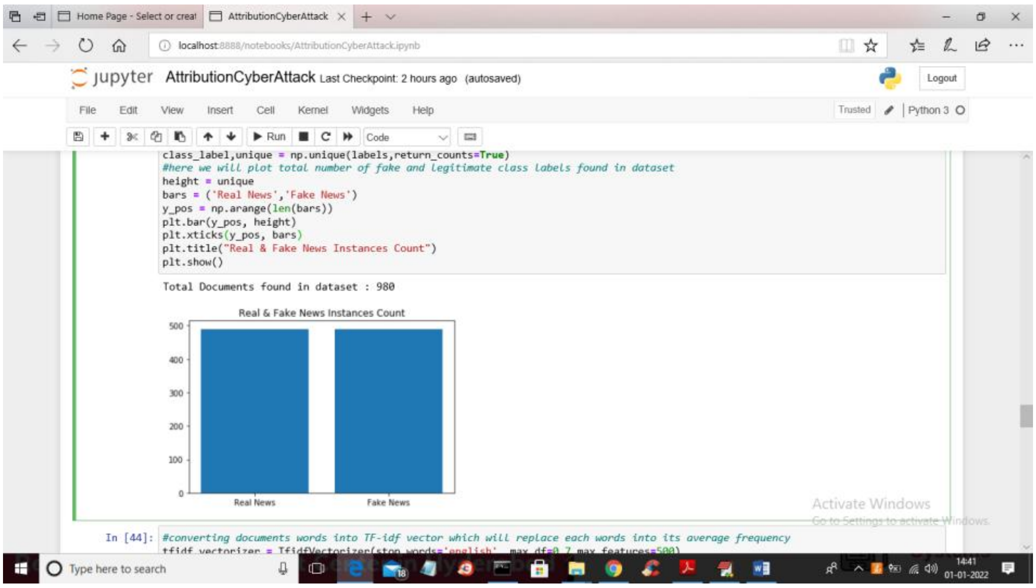


Home Page - Select or creat
AttributionCyberAttack
localhost:8888/notebooks/AttributionCyberAttack.ipynb
jupyter AttributionCyberAttack Last Checkpoint: 2 hours ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Activate Windows
Go to Settings to activate Windows.
Type here to search
14:39
01-01-2022

```
In [51]: #check if documents and lables already process or not and if not process then read all documents and then converted to
         #numpy array and if documents already processed then load it
         documents = []
         labels = []
         if os.path.exists('model/documents.npy'):
             documents = np.load("model/documents.npy",allow_pickle=True)
             labels = np.load("model/labels.npy")
         else:
             news_fake = "Dataset/fakeNewsDataset/fake"
             news_legit = "Dataset/fakeNewsDataset/legit"
             iterateDocument(news_fake)
             iterateDocument(news_legit)
             documents = np.asarray(documents)
             labels = np.asarray(labels)
             np.save('model/documents',documents)
             np.save('model/labels',labels)
         print("Total Documents found in dataset : "+str(documents.shape[0]))
         class_label,unique = np.unique(labels,return_counts=True)
         #here we will plot total number of fake and legitimate class labels found in dataset
         height = unique
         bars = ('Real News','Fake News')
         y_pos = np.arange(len(bars))
         plt.bar(y_pos, height)
         plt.xticks(y_pos, bars)
         plt.title("Real & Fake News Instances Count")
         plt.show()

         Total Documents found in dataset : 980
```

Fig.14

In above figure we are loading FAKE and LEGIT (real) news document and in blue colour text we can see total 980 news documents are loaded and in below figure we can see graph of fake and real news



```
class_label,unique = np.unique(labels,return_counts=True)
#here we will plot total number of fake and legitimate class labels found in dataset
height = unique
bars = ('Real News','Fake News')
y_pos = np.arange(len(bars))
plt.bar(y_pos, height)
plt.xticks(y_pos, bars)
plt.title("Real & Fake News Instances Count")
plt.show()
```

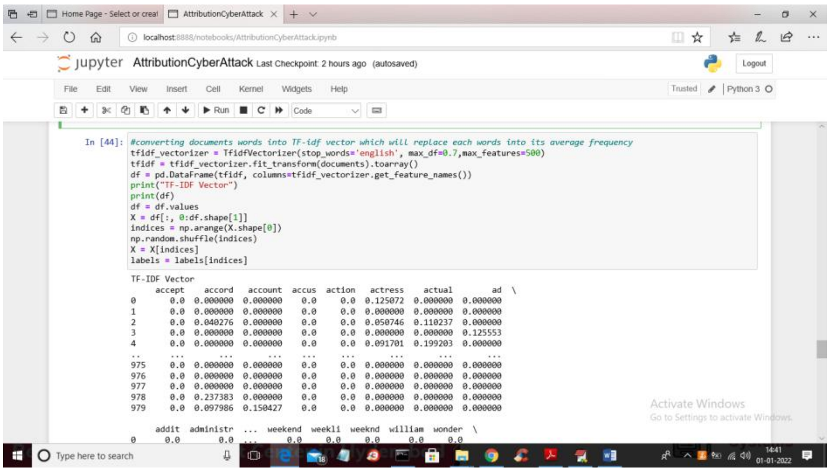
```
Total Documents found in dataset : 980
```

jupyter AttributionCyberAttack Last Checkpoint: 2 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Real & Fake News Instances Count
500
400
300
200
100
0
Real News
Fake News
Activate Windows
Go to Settings to activate Windows.
Type here to search
14:41
01-01-2022

```
In [44]: #converting documents words into TF-idf vector which will replace each words into its average frequency
```

Fig.15

In below figure we are converting dataset into TF-IDF vector which will contains average frequency of each word and this vector will be trained with random forest



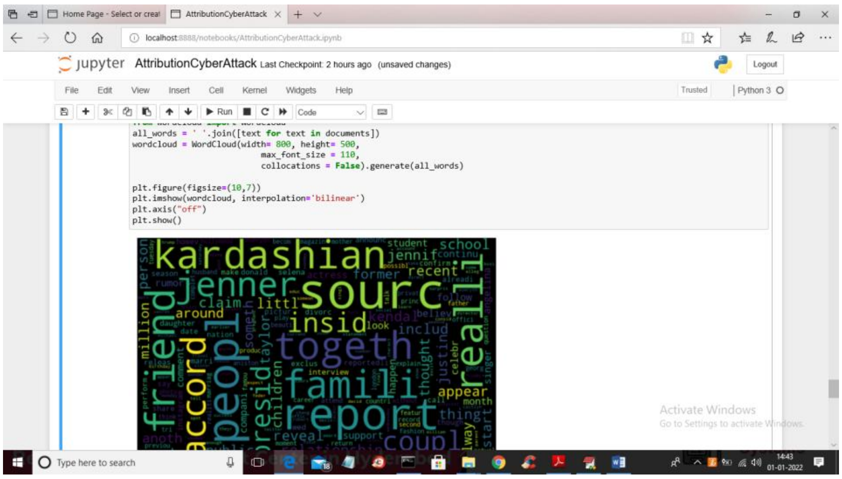
```
In [44]: #converting documents words into TF-idf vector which will replace each words into its average frequency
tfidf_vectorizer = TfidfVectorizer(stop_words='english', max_df=0.7,max_features=500)
tfidf = tfidf_vectorizer.fit_transform(documents).toarray()
df = pd.DataFrame(tfidf, columns=tfidf_vectorizer.get_feature_names())
print("TF-IDF Vector")
print(df)
df = df.values
X = df[:, 0:df.shape[1]]
indices = np.arange(X.shape[0])
np.random.shuffle(indices)
X = X[indices]
labels = labels[indices]
```

```
TF-IDF Vector
     accept    accord   account  accus  action   actress    actual        ad  \
0       0.0  0.000000  0.000000    0.0     0.0  0.125072  0.000000  0.000000
1       0.0  0.000000  0.000000    0.0     0.0  0.000000  0.000000  0.000000
2       0.0  0.040276  0.000000    0.0     0.0  0.050746  0.110237  0.000000
3       0.0  0.000000  0.000000    0.0     0.0  0.000000  0.000000  0.125553
4       0.0  0.000000  0.000000    0.0     0.0  0.091701  0.199203  0.000000
..      ...       ...       ...    ...     ...       ...       ...       ...
975     0.0  0.000000  0.000000    0.0     0.0  0.000000  0.000000  0.000000
976     0.0  0.000000  0.000000    0.0     0.0  0.000000  0.000000  0.000000
977     0.0  0.000000  0.000000    0.0     0.0  0.000000  0.000000  0.000000
978     0.0  0.237383  0.000000    0.0     0.0  0.000000  0.000000  0.000000
979     0.0  0.097986  0.150427    0.0     0.0  0.000000  0.000000  0.000000

     addit  administr  ...  weekend  weekli  weeknd  william  wonder  \
0      0.0        0.0  ...      0.0     0.0     0.0      0.0     0.0
```

jupyter AttributionCyberAttack Last Checkpoint: 2 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Activate Windows
Go to Settings to activate Windows.
Type here to search
14:41
01-01-2022

Fig.16

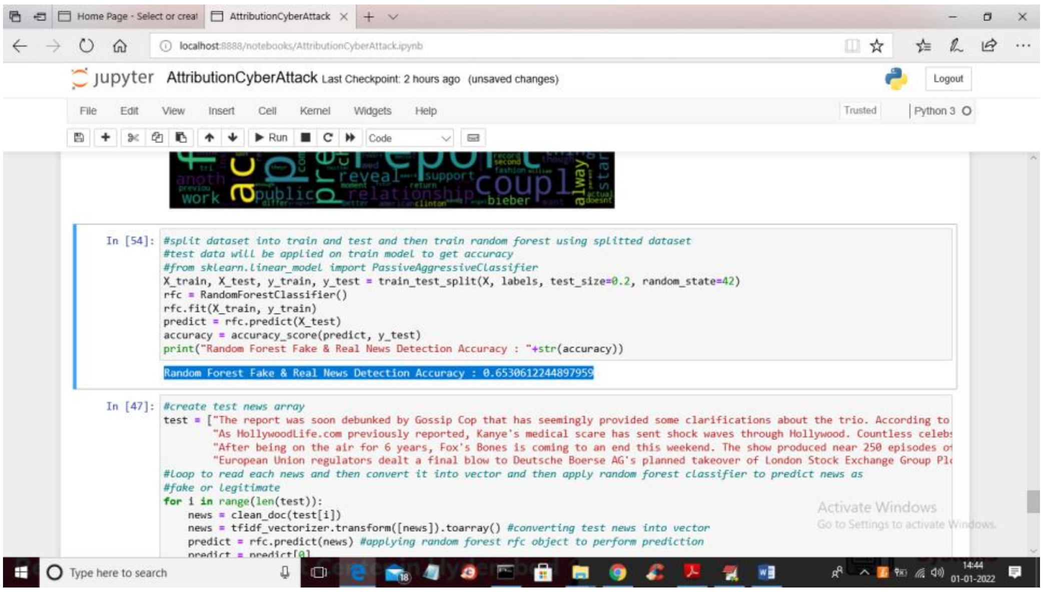
In above figure first line contains WORD name and below its we can find its frequency and if word not available then 0 will put. Now in below figure we are displaying WORD cloud to know which words are appearing most



```
all_words = ' '.join([text for text in documents])
wordcloud = WordCloud(width= 800, height= 500,
                      max_font_size = 110,
                      collocations = False).generate(all_words)

plt.figure(figsize=(10,7))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```

Fig.17

In above word cloud if word occur more time then its size will be huge else small. In below figure I am training frequency vector with random forest

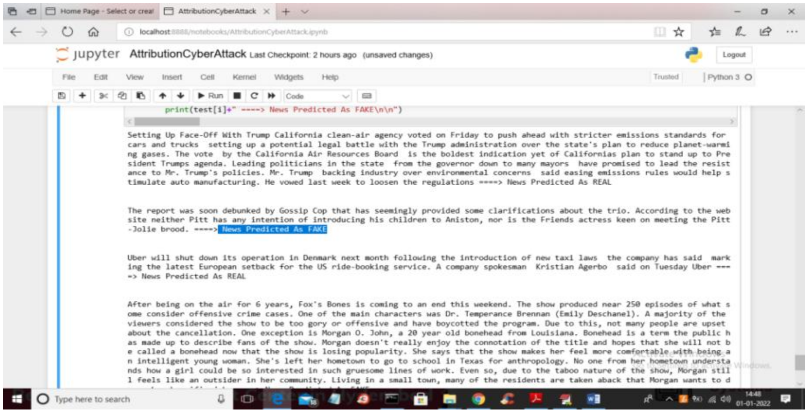


```
In [54]: #split dataset into train and test and then train random forest using splitted dataset
         #test data will be applied on train model to get accuracy
         #from sklearn.linear_model import PassiveAggressiveClassifier
         X_train, X_test, y_train, y_test = train_test_split(X, labels, test_size=0.2, random_state=42)
         rfc = RandomForestClassifier()
         rfc.fit(X_train, y_train)
         predict = rfc.predict(X_test)
         accuracy = accuracy_score(predict, y_test)
         print("Random Forest Fake & Real News Detection Accuracy : "+str(accuracy))

         Random Forest Fake & Real News Detection Accuracy : 0.6530612244897959

In [47]: #create test news array
         test = ["The report was soon debunked by Gossip Cop that has seemingly provided some clarifications about the trio. According to
                 "As HollywoodLife.com previously reported, Kanye's medical scare has sent shock waves through Hollywood. Countless celeb:
                 "After being on the air for 6 years, Fox's Bones is coming to an end this weekend. The show produced near 250 episodes of
                 "European Union regulators dealt a final blow to Deutsche Boerse AG's planned takeover of London Stock Exchange Group Pl
         #loop to read each news and then convert it into vector and then apply random forest classifier to predict news as
         #fake or legitimate
         for i in range(len(test)):
             news = clean_doc(test[i])
             news = tfidf_vectorizer.transform([news]).toarray() #converting test news into vector
             predict = rfc.predict(news) #applying random forest rfc object to perform prediction
```

jupyter AttributionCyberAttack Last Checkpoint: 2 hours ago (unsaved changes)
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code
Logout

Fig.18

In above figure with random forest we got FAKE news detection accuracy as 65% and in below figure we are applying same trained model on new test data to predict it as FAKE or REAL. In below figure we are giving some sentences and then random forest will predict it as REAL or FAKE



AttributionCyberAttack Last Checkpoint: 2 hours ago (unsaved changes)

```
print(test[i]+" ---->  News Predicted As FAKE\n\n")
```

```
Setting Up Face-Off With Trump California clean-air agency voted on Friday to push ahead with stricter emissions standards for
cars and trucks  setting up a potential legal battle with the Trump administration over the state's plan to reduce planet-warmi
ng gases. The vote  by the California Air Resources Board  is the boldest indication yet of Californias plan to stand up to Pre
sident Trumps agenda. Leading politicians in the state  from the governor down to many mayors  have promised to lead the resist
ance to Mr. Trump's policies. Mr. Trump  backing industry over environmental concerns  said easing emissions rules would help s
timulate auto manufacturing. He vowed last week to loosen the regulations ====> News Predicted As REAL


The report was soon debunked by Gossip Cop that has seemingly provided some clarifications about the trio. According to the web
site neither Pitt has any intention of introducing his children to Aniston, nor is the Friends actress keen on meeting the Pitt
-Jolie brood. ====> News Predicted As FAKE


Uber will shut down its operation in Denmark next month following the introduction of new taxi laws  the company has said  mark
ing the latest European setback for the US ride-booking service. A company spokesman  Kristian Agerbo  said on Tuesday Uber ---
-> News Predicted As REAL


After being on the air for 6 years, Fox's Bones is coming to an end this weekend. The show produced near 250 episodes of what s
ome consider offensive crime cases. One of the main characters was Dr. Temperance Brennan (Emily Deschanel). A majority of the
viewers considered the show to be too gory or offensive and have boycotted the program. Due to this, not many people are upset
about the cancellation. One exception is Morgan O. John, a 20 year old bonehead from Louisiana. Bonehead is a term the public h
as made up to describe fans of the show. Morgan doesn't really enjoy the connotation of the title and hopes that she will not b
e called a bonehead now that the show is losing popularity. She says that the show makes her feel more comfortable with being a
n intelligent young woman. She's left her hometown to go to school in Texas for anthropology. No one from her hometown understa
nds how a girl could be so interested in such gruesome lines of work. Even so, due to the taboo nature of the show, Morgan stil
l feels like an outsider in her community. Living in a small town, many of the residents are taken aback that Morgan wants to d
```

Fig.19

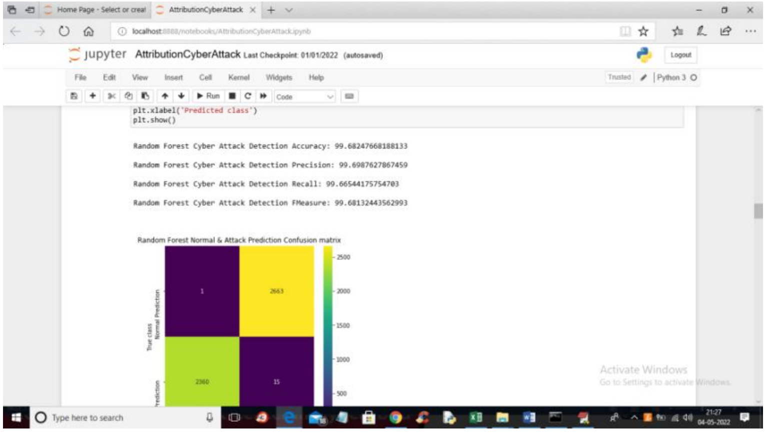
In above figure after sentence we are displaying predicted output as FAKE or REAL after =➔ arrow symbol

Extension Outcomes

In propose paper author has used Random Forest algorithm to detect cyber-attack from Intrusion dataset and then detecting FAKE news by using FAKE dataset. Author has used traditional Random Forest algorithm so as extension we have used combination of 3 different latest classifiers such as AdaBoost, XGBoost and Bagging classifier to make hybrid ensemble algorithm with the help of VotingClassifier.

Voting classifier takes multiple classifier algorithms as input and then train all those algorithms and then voted out or select algorithm with best accuracy. So by applying this algorithm on both Intrusion and Fake news giving better accuracy compare to Random Forest.

In below figure I am showing Random Forest Performance output



AttributionCyberAttack Last Checkpoint: 01/01/2022 (autosaved)

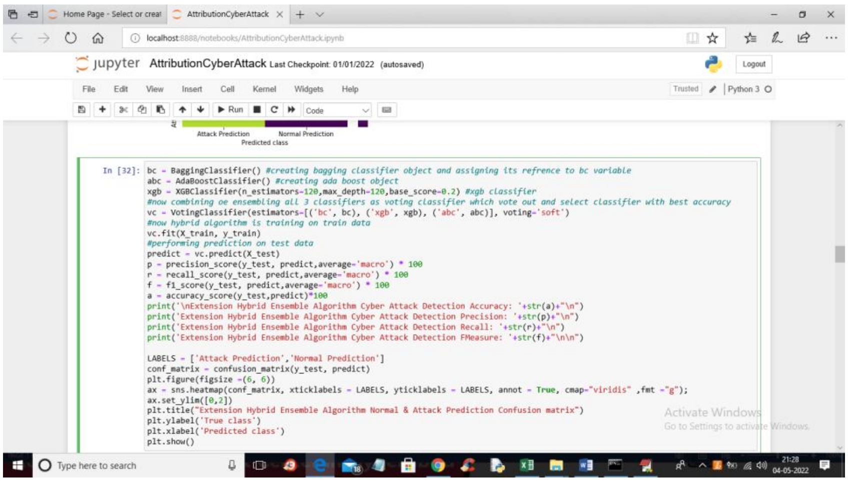
```
plt.xlabel('Predicted class')
plt.show()
```

```
Random Forest Cyber Attack Detection Accuracy: 99.68247668188133

Random Forest Cyber Attack Detection Precision: 99.6987627867459

Random Forest Cyber Attack Detection Recall: 99.66544175754703

Random Forest Cyber Attack Detection FMeasure: 99.68132443562993
```

Random Forest Normal & Attack Prediction Confusion matrix
1
2663
2360
15
2500
2000
1500
1000
500
True class
Normal Prediction
Prediction

Fig.20

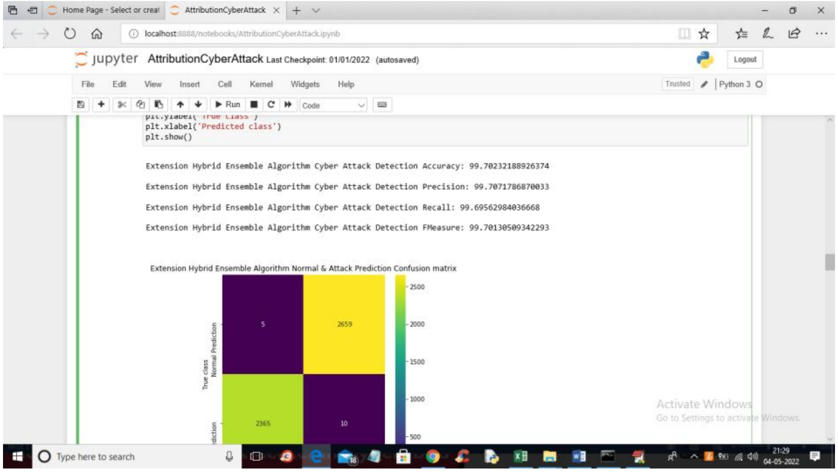
In above figure with Random Forest we got 99.68% accuracy and in confusion matrix graph we can see Random forest wrongly predict 15 and 1 records so total 16 records wrongly predicted. Below is the output of extension hybrid algorithm on same dataset



```
In [32]: bc = BaggingClassifier() #creating bagging classifier object and assigning its refrence to bc variable
         abc = AdaBoostClassifier() #creating ada boost object
         xgb = XGBClassifier(n_estimators=120,max_depth=120,base_score=0.2) #xgb classifier
         #now combining oe ensembling all 3 classifiers as voting classifier which vote out and select classifier with best accuracy
         vc = VotingClassifier(estimators=[('bc', bc), ('xgb', xgb), ('abc', abc)], voting='soft')
         #now hybrid algorithm is training on train data
         vc.fit(X_train, y_train)
         #performing prediction on test data
         predict = vc.predict(X_test)
         p = precision_score(y_test, predict,average='macro') * 100
         r = recall_score(y_test, predict,average='macro') * 100
         f = f1_score(y_test, predict,average='macro') * 100
         a = accuracy_score(y_test,predict)*100
         print('\nExtension Hybrid Ensemble Algorithm Cyber Attack Detection Accuracy: '+str(a)+"\n")
         print('Extension Hybrid Ensemble Algorithm Cyber Attack Detection Precision: '+str(p)+"\n")
         print('Extension Hybrid Ensemble Algorithm Cyber Attack Detection Recall: '+str(r)+"\n")
         print('Extension Hybrid Ensemble Algorithm Cyber Attack Detection FMeasure: '+str(f)+"\n\n")

         LABELS = ['Attack Prediction','Normal Prediction']
         conf_matrix = confusion_matrix(y_test, predict)
         plt.figure(figsize =(6, 6))
         ax = sns.heatmap(conf_matrix, xticklabels = LABELS, yticklabels = LABELS, annot = True, cmap="viridis" ,fmt ="g");
         ax.set_ylim([0,2])
         plt.title("Extension Hybrid Ensemble Algorithm Normal & Attack Prediction Confusion matrix")
         plt.ylabel('True class')
         plt.xlabel('Predicted class')
         plt.show()
```

Fig.21

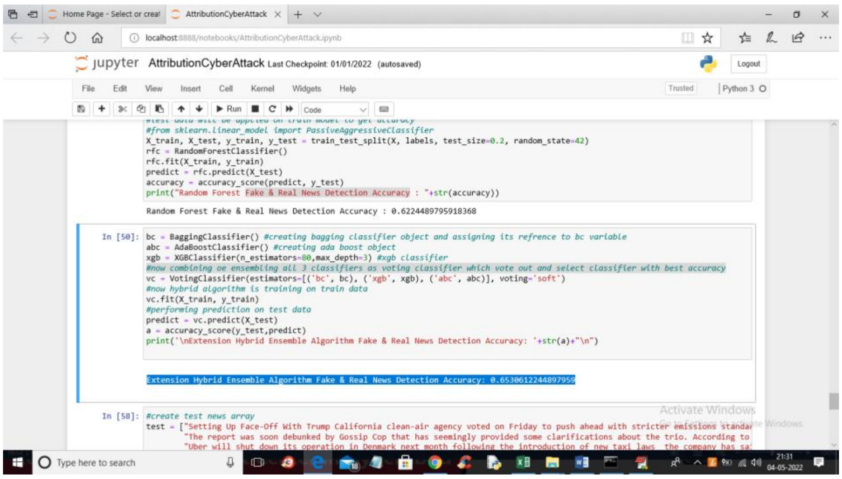
In above figure you can see Extension hybrid algorithm which is combining 3 different classifiers and in above figure read light blue colour comments to know about extension algorithm and below is the output


Home Page - Select or creat
AttributionCyberAttack
localhost:8888/notebooks/AttributionCyberAttack.ipynb
jupyter AttributionCyberAttack Last Checkpoint: 01/01/2022 (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code
Extension Hybrid Ensemble Algorithm Normal & Attack Prediction Confusion matrix
5
2659
2365
10
2500
2000
1500
1000
500
True class
Normal Prediction
diction
Activate Windows
Go to Settings to activate Windows.
Type here to search
21:29
04-05-2022

```
plt.ylabel('True class')
plt.xlabel('Predicted class')
plt.show()

Extension Hybrid Ensemble Algorithm Cyber Attack Detection Accuracy: 99.70232188926374

Extension Hybrid Ensemble Algorithm Cyber Attack Detection Precision: 99.7071786870033

Extension Hybrid Ensemble Algorithm Cyber Attack Detection Recall: 99.69562984036668

Extension Hybrid Ensemble Algorithm Cyber Attack Detection FMeasure: 99.70130509342293
```

Fig.22

In above figure with extension algorithm we got 99.70% accuracy and Random Forest was 99.68 and in above confusion matrix extension algorithm wrongly predict 10 and 5 records so total 15 records wrongly predicted by extension algorithm and random forest wrong prediction was 16 so extension is better than Random Forest. Below is the Random Forest accuracy of Fake News dataset



```
#test data will be applied on train model to get accuracy
#from sklearn.linear_model import PassiveAggressiveClassifier
X_train, X_test, y_train, y_test = train_test_split(X, labels, test_size=0.2, random_state=42)
rfc = RandomForestClassifier()
rfc.fit(X_train, y_train)
predict = rfc.predict(X_test)
accuracy = accuracy_score(predict, y_test)
print("Random Forest Fake & Real News Detection Accuracy : "+str(accuracy))

Random Forest Fake & Real News Detection Accuracy : 0.6224489795918368

In [50]: bc = BaggingClassifier() #creating bagging classifier object and assigning its refrence to bc variable
         abc = AdaBoostClassifier() #creating ada boost object
         xgb = XGBClassifier(n_estimators=80,max_depth=3) #xgb classifier
         #now combining oe ensembling all 3 classifiers as voting classifier which vote out and select classifier with best accuracy
         vc = VotingClassifier(estimators=[('bc', bc), ('xgb', xgb), ('abc', abc)], voting='soft')
         #now hybrid algorithm is training on train data
         vc.fit(X_train, y_train)
         #performing prediction on test data
         predict = vc.predict(X_test)
         a = accuracy_score(y_test,predict)
         print('\nExtension Hybrid Ensemble Algorithm Fake & Real News Detection Accuracy: '+str(a)+"\n")


         Extension Hybrid Ensemble Algorithm Fake & Real News Detection Accuracy: 0.6530612244897959

In [58]: #create test news array
         test = ["Setting Up Face-Off With Trump California clean-air agency voted on Friday to push ahead with stricter emissions standar
                 "The report was soon debunked by Gossip Cop that has seemingly provided some clarifications about the trio. According to
                 "Uber will shut down its operation in Denmark next month following the introduction of new taxi laws  the company has sa
```

Fig.23

In the figure above, Random Forest got 0.62 percent accuracy, and the extension algorithm got 0.65 percent accuracy, which you can see in blue text. So, the extension algorithm is more accurate than the proposed random algorithm for both intrusion data and fake news data.

CONCLUSION

Cyber attacks are a big research problem that is important for the Internet's stability and continued existence. Even though the Internet has become an important part of the world's social and economic life, it is also becoming a place where fights are likely to happen. Nation states are using the Internet as a strategic way to make up for differences in their militaries. They prefer to use cyber attacks instead of physical attacks to show they don't like the policies of another country. Because of how anonymous the Internet is, countries can say they didn't do it. Without clear attribution, it's hard to find the culprits or use international law to hold them accountable. Recent changes have made it easier to figure out who did something on the Internet. This is done through a complicated and time-consuming process of forensics on different pieces of evidence, such as code provenance, network data, and ip/domain addresses. Intelligence agencies have gathered information about what hackers, organisations, and nation states have done online in the past. This helps them figure out where the malware came from. Also, possible bad actors can be identified with a fair amount of certainty by looking at their means, motives, and opportunities. More recently, social bots have become a major threat to national stability. At the moment, social bots are found by analysing message statistics (like how often they post, how long it takes them to respond, etc.) and linguistics. Attribution, however, is still done by hand. Most of the work of filtering out fake news is still done by hand, and social bots have a long way to go before they can reliably spot fake news. Technical analysis and correlation with human intelligence are used to find and identify attacks. There is still a lot of work to do to standardise forensic processes, figure out the right data to collect, improve and standardise analysis techniques, and correlate these with intelligence from the ground.

Bibilography

1. B. M. Leiner et al., “A brief history of the Internet,” ACM SIGCOMM Comput. Commun. Rev., vol. 39, no. 5, pp. 22–23, 1997.
2. H. L. Armstrong and P. J. Forde, “Forde Internet anonymity practices in computer crime,” Inf. Manag. Comput. Security, vol. 115, no. 5, pp. 209–215, 2003.
3. B. Warf, “Geographies of global Internet censorship,” GeoJournal, vol. 76, no. 1, pp. 1–23, 2011.
4. B. Krekel, P. Adams, and G. Bakos, “Occupying the information high ground: Chinese capabilities for computer network operations and cyber espionage,” Int. J. Comput. Res., vol. 21, no. 4, p. 333, 2014.
5. N. J. Shallcross, “Social media and information operations in the 21st century,” J. Inf. Warfare, vol. 16, no. 1, pp. 1–12, 2017.
6. Weedon, W. Nuland, and A. Stamos. (2017). Information Operations and Facebook. Facebook.Online Available:https://www.mm.dk/wpcontent/uploads/2017/05/facebook-and-information-operations-v1.pdf
7. S. W. Brenner, “At light speed: Attribution and response to cybercrime/terrorism/warfare,” J. Crim. Law Criminol., vol. 97, p. 379, Mar. 2007.
8. H. S. Lin “Attribution of malicious cyber incidents: From soup to nuts,” Legal Perspectives Inf. Syst. J., to be published.
9. E. M. Mudrinich, “Cyber 3.0: The department of defense strategy for operating in cyberspace and the attribution problem,” AFL Rev., vol. 68, p. 167, Jul. 2012
10. B. Edwards, A. Furnas, S. Forrest, and R. Axelrod, “Strategic aspects of cyberattack, attribution, and blame,” Proc. Nat. Acad. Sci. USA, vol. 114, no. 11, pp. 2825–2830, 2017.

11. H. Berghel, “On the problem of (cyber) attribution,” IEEE Comput., vol. 50, no. 3, pp. 84–

12. F. J. Egloff, “Public attribution of cyber intrusions,” J. Cybersecurity, vol. 6, no. 1, 2020, Art. no. tyaa012.

13. F. J. Egloff and M. Smeets, “Publicly attributing cyber attacks: a framework,” J. Strategic Stud., to be published.

14. M. Mueller, K. Grindal, B. Kuerbis, and F. Badiei, “Cyber attribution,” Cyber Defense Rev., vol. 4, no. 1, pp. 107–122, 2019.

15. T. Rid and B. Buchanan, “Attributing cyber attacks,” J. Strategic Stud., vol. 38, nos. 1–2, pp. 4–37, 2015.

16. J. Healey, Beyond Attribution: A Vocabulary for National Responsibility for Cyber Attacks. Vienna, VA, USA: Cyber Conflict Stud. Assoc., 2010.

17. J. Canfil, “Honing cyber attribution: A framework for assessing foreign state complicity,” J. Int. Affairs, vol. 70, no. 1, pp. 217–226, 2016. [Online]. Available: https://www.jstor.org/stable/90012607

18. D. Wheeler and G. Larsen, Techniques for Cyber Attack Attribution, Inst. Defense Anal., Alexandria, VA, USA, 2003

19. N. Tsagourias, “Cyber attacks, self-defence and the problem of attribution,” J. Conflict Security Law, vol. 17, no. 2, pp. 229–244, 2012.

20. K. E. Eichensehr, “The law and politics of cyberattack attribution,” UCLA Law Rev., vol. 67, p. 520, Jan. 2020