DNS OVER HTTPS: DETECTING MALICIOUS PACKETS USING MACHINE LEARNING

A DISSERTATION

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE
OF

MASTER OF TECHNOLOGY

IN

INFORMATION SYSTEMS

Submitted By:

SAURAV KUMAR

(2K20/ISY/19)

Under the supervision of

PROF. KAPIL SHARMA

HOD, DEPT OF IT

DEPARTMENT OF INFORMATION TECHNOLOGY

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

MAY, 2022

CANDIDATE'S DECLARATION

I, SAURAV KUMAR, Roll No. 2k20/ISY/19 student of M.Tech Information Systems, hereby declare that the project Dissertation titled "DNS overs HTTPS: Detecting malicious packets using machine learning" which is submitted by me to the Department of information technology, Delhi Technological University, Delhi in partial fulfillment of the required for the award of the degree of Master of Technology is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma Associateship, Fellowship, or other similar titles or recognition.

Place: Delhi Saurav Kumar

Date: 23 May

CERTIFICATE

I hereby certify that the Project Dissertation titled “DNS overs HTTPS: Detecting malicious packets using machine learning” which is submitted by Saurav Kumar, Roll No. 2K20/ISY/19, Information Systems, Delhi Technological University, Delhi in partial fulfillment of the requirements for the award of the degree of Masters of Technology, is a record of the project work carried out by the students under my supervision. To the best of my knowledge this work has not been submitted in the part or full for any Degree or Diploma to this university or elsewhere

Place: Delhi
Date

Prof Kapil Sharma
HOD, DEPT OF IT

ACKNOWLEDGEMENT

I express my gratitude to my major project guide Prof. Kapil Sharma, HOD Department of IT, Delhi Technological University, for the valuable support and guidance he provided in making this major project. It is my pleasure to record my sincere thanks to my respected guide for his constructive criticism and insight without which the project would not have been shaped as it has.

I humbly extend my words of gratitude to other faculty members of this department for providing their valuable help and time whenever it was required.

Saurav Kumar

2K20/ISY/19

M.TECH

INFORMATION SYSTEMS

ABSTRACT

With the implementation of the HTTPs, the communication between the website and the browser has become secure. The HTTPs uses the TSL encryption, which uses the asymmetric key to encrypt the data being exchanged. Https is only responsible for data transfer, the IP translation is done by the DNS. DNS was first introduced in 1983, since then it has been responsible for resolving the URL into the IP address. A non-cached DNS query is resolved by first sending it to the Recursive DNS Servers, which is then sent to the Authoritative DNS Servers which finally returns the IP address. All these transmissions are not encrypted and can be read by any on-route router, this compromised the privacy of the users and they became susceptible to tracking and spoofing. In 2018, DNS over HTTPS was introduced and it greatly enhanced the privacy of the users by encrypting the query. DoH works by sending the DNS query over https directly over to the DoH server thus eliminating the information leakage. DoH solved the privacy issues but it resulted in other problems. DNS data is actively used by intrusion detection systems, firewalls and by various corporations to either block, control or to monitor the traffic before it enters into the network. But now as the DNS data would be encrypted thus

these services can't be carried out properly. It can even lead to serious consequences if a malware bypasses the firewall. Due to all these reasons, it hasn’t been welcomed by many corporations. But Google and Firefox have already decided to provide this feature in their browsers. Thus, the network needs to be monitored based on other properties to filter the malicious traffic. In this project we have used artificial neural network to train the model. The model has shown exceptional accuracy. Our work is different from all the previous work

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATION

LIST OF ABBREVIATION

IP: INTERNET PROTOCOL

DOH: DNS OVER HTTPS

DNS: DOMAIN NAME SYSTEM

KNN: K NEAREST NEIGHBOUR

ANN: ARTIFICIAL NEURAL NETWORK

HTTPS: HYPERTEXT TRANSFER PROTOCOL

CHAPTER 1

INTRODUCTION

1.1DOMAIN NAME SYSTEM

The Domain Name System (DNS) is an effective mechanism for the operation of the Internet. Essentially, it is responsible for mapping human-readable Internet destinations to given IP addresses. In other words, machines making up the Internet are addressed using IP addresses. However, Internet users usually do not desire to remember these IP addresses; instead, they use them to guide their traffic. Accordingly, users depend on domain names to direct their traffic, making it more straightforward for them to remember. It is also a seamless method of advertising to influence online users to visit specific websites. To this end, employing DNS servers enables client computers to establish the IP address associated with a particular domain requested by the Internet user.

1.2 ISSUES WITH DNS

While DNS encryption is considered beneficial as a new trend, it is also associated with some potential issues, including increased centralization, a vital aspect of the Internet infrastructure. Essentially the DNS infrastructure is less resistant to interruption caused by configuration errors, intrusion, and manipulation.[1] These dangers are not merely hypothetical: In 2016, a cyberattack on DNS infrastructure left several websites inaccessible. DNS requests are easily manipulated, culminating in information control and restriction. DNS vulnerabilities are also somewhat prevalent from a long time.[2]

1.3 DNS OVER HTTPS (DOH)

DNS over HTTPS (DoH) is a protocol intended to address this privacy issue[1]. Its premise is relatively simple: instead of sending DNS queries and replies out in cleartext, they will be sent out wrapped in an HTTPS GET or POST request, depending on the browser. Because HTTPS is encrypted and authenticated via TLS, an attacker's ability to read or change them is significantly reduced through this process.

DoH is not widely used and is still regarded as "in development," this does not rule out the possibility that it is currently accessible. The transition from regular DNS to DoH is a significant milestone. [3]It has been decades since DNS traffic on the Internet has been sent in an unencrypted fashion. As a result, many DNS servers may be unable to create and send encrypted answers unless they undergo an extensive update. Consequently, some sites may be inaccessible while utilizing the DoH due to this restriction. As the name implies, this essentially wraps DNS communication inside HTTPS transmission, ensuring that DNS queries and replies cannot be intercepted or altered in cleartext. Thus, using DoH assures an improved performance over the traditional DNS.

The discussion on the influence of DoH on the corporate sector is a trending topic regarding policies in the Internet ecosystem.[1] DoH has been an issue for businesses since it was first suggested. It offers a means to override centrally enforced DNS settings, allowing workers to utilise DoH to avoid DNS-based traffic filtering systems. Because today's DNS servers do not accept DoH queries, programs now supporting DoH are mainly hardcoded DoH servers.[4] System administrators must monitor such systems to combat attempts of attacks. The question of how to accomplish such milestones necessitates implementing policies that guide such actions; thus, the increasing need for corporate- and user-level policies to ensure security.

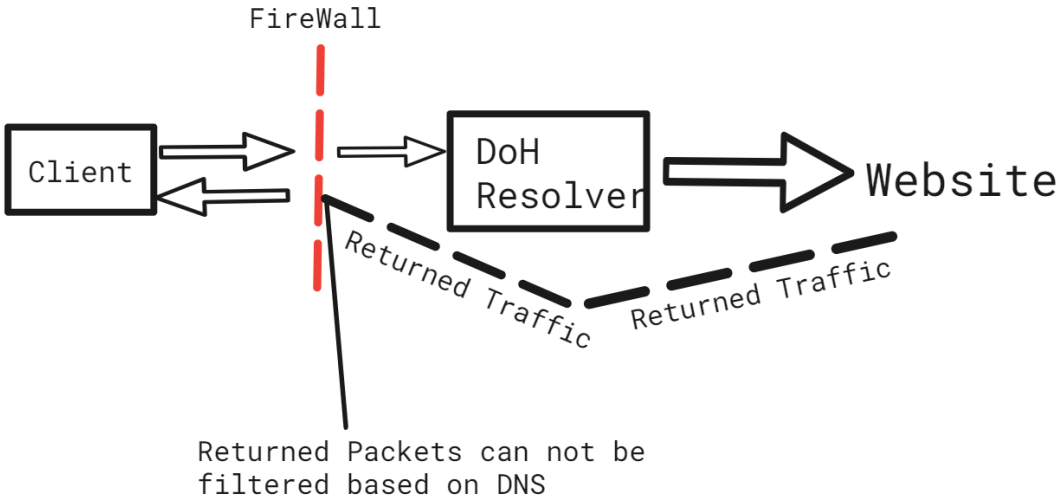

Fig 1.1 Issues raised due to DoH

1.4 DISTINCTIONS BETWEEN WEB TRAFFIC AND DOH TRAFFIC

Even though the Doh traffic appears to be similar to the normal web traffic but there are many differences between the two. The first major difference between them is the number of packets involved in each response. [5]In case of normal web traffic there are less packets involved but the DoH communication uses a large number of packets each time.[5] There are at least 5 packets used in each time.

The next difference between the normal traffic and Doh traffic is the flow duration of the packets. [5]In case of DoH the flow durations are usually always longer than usual web traffic because it uses multiple packets in each response.

The next difference is the difference in the packet sizes of normal web traffic and DoH traffic. This difference can easily be used to determine the difference between the two.[5]

At present DoH can be used in two ways; first way is the one in which it can be used from the browser directly by enabling it in the browser settings. Second method to use the DoH is to route the DNS queries through a central DOH proxy. This proxy performs a special task of changing the format of these DNS queries into the DoH format.

1.5 WORKING OF DOMAIN NAME SYSTEM

Humans are used to entering the human readable URL in order to reach to a website. But at the root level these searches are not performed using these URLs. It is very difficult to remember the numeric IP address of the website. That is why a mechanism was required which could take care of the numeric IP address for respective URLs. In 1983 DNs was introduced by Paul Mockapetris. DNS is responsible for converting the human readable URLs into machine readable IP address.

1.6 DNS RESOLUTION

Resolution of URLs into IP address takes place in 8 steps

- In the first step the website generates the query from the user end and that query is sent to the recursive resolver.
- In the second step the recursive resolver is further responsible for moving the query forward. This query is sent to the root server.
- In the third step, the query which was sent to the root server generates an IP address of the TLD (Top level domain). This IP address is then sent to the recursive resolver
- In fourth step the recursive resolver further queries with the TLD(Top level domain) server
- In Fifth step TDL (Top level domain) returns the address of the domain name server
- In Sixth step the query is forwarded to the domain name server.
- In Seventh step the IP address is returned by the domain name server
- In the last step returned IP address is sent back to the web browser

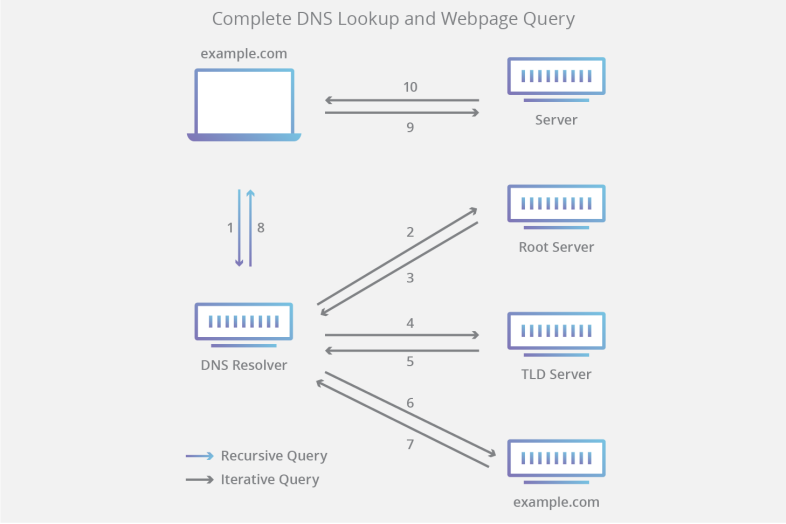

Fig 1.2 DNS resolution

1.7 WORKING OF HTTPS

HTTPS uses the TSL encryption protocol to encrypt the communication taking place. TSL uses the concept of asymmetric key cryptography. [6]In asymmetric cryptography a set of public and private key is generated for each website.

The private key is held by the owner of the website and the public key is available to all those who wants to communicate with the website. Once the messages have been encrypted using the private key, only public key can be used to decrypt them and vice versa

1.8 WORKING OF DOH

DNS over https is a very new protocol, it was introduced in 2018. It is implemented using two approaches.

- The first approach uses the DNS Wireformat; it uses the same format as that of a DNS query but It wraps the DNS into HTTPS data using POST and Get method.
- The second approach is used by Google; it uses HTTPS GET based on Json messaging

Though both of these approaches are supported but Wireformat messages are being currently used by all the browsers. HTTPSs2 is the protocol recommended for DOH. This protocol starts with a handshake; which is then followed by the connection preface. After the handshake has been completed the request and response takes place.

At present the doh service is provided by limited servers. Even though the DNS queries are encrypted but they must be decrypted in order to be translated into IPs. In case of doh the entire process of name resolution i.e. converting the URLs into IP, is moved behind the server providing the DOH service. The encrypted transmission takes place with the DoH server, the server then places a DNS query on its behalf by requesting the resolver with its own ip to interact with the

conventional naming servers. Some servers even use the query name minimization to improve the privacy

1.9 SUPERVISED MACHINE LEARNING

Supervised learning algorithms can easily be implemented on labeled data. Few of the Types of supervised algorithms are KNN, Random Forest, Decision Tree. These algorithms have been implements in previous works.

1.9.1 K-NEAREST NEIGHBOR

It is a supervised machine learning algorithm which is used for classification. This algorithm considers that similar things can be found in closeness to each other. Based on this closeness it calculates the probability of data point belonging to a group. In case of KNN, during the training phase data is simply stored and no calculations are carried out thus there is no training performed at this phase. Due to this reason this algo is also called a lazy learning algorithm In case of KNN no presumptions are made about the data, that is why it is considered to as non-parametric method.[7]

Algorithms of KNN

- Load the data.
- Set the K value to any arbitrary value.
- For each point in data:
- Calculate Euclidean distance between the current data point and other data points.

$$d(\mathbf{p},\mathbf{q}) = d(\mathbf{q},\mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2} = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2}. \quad (1.1)$$

- Add the calculated distance and the index to an ordered list.
- Sort the list and select first K entries.
- Label the selected k points and return the mode.

1.9.2 DECISION TREE

This algorithm falls under the supervised technique. It works both as classification and regression technique. These are of two types depending on the target variable.

- **Categorical Variable Tree** a tree working on the categorical variable is called categorical.
- **Continuous Variable tree**: a tree which has the target variable as the continuous variable is called a continuous tree.

Certain assumptions are to be made while creating a decision tree.

- Whole training data is considered as root to start with
- Continuous values are represented approximately using the discreet value.
- Recursive distribution of records is carried out based on the random value
- Statistical approach is used in internal root ordering.

Algorithm for tree construction

- Start with the dataset as root node S
- Use ASM for find the best attribute
- Split the root S into subsets which contains the best value
- Repeat step 3 to create the trees until it can’t be done any further

1.9.3 C4.5 TREE

This tree is the modification of ID3. It uses Gain ratio as the goodness function. Gain ration minimizes the overfitting which occurred in ID3

Gain ratio

As the IG suffered from overfitting due to its preference for features with multiple categories. Gain ratio minimized this by penalizing those features. It uses Intrinsic information formula to do so.

$$-\sum_{i=1}^{c} P(x_i) log_b P(x_i) \qquad (1.2)$$

$$Gain = Entropy(before) - \sum_{j=1}^{K} Entropy(j, after) \qquad (1.3)$$

$$\text{GainRatio(A)} = \frac{\text{Gain(A)}}{\text{SplitInfo(A)}} \qquad (1.4)$$

$$\text{Split Information} = \sum -\text{ratio} \times \log_2 \text{ratio} \; \forall \; \text{category} \in \text{feature} \qquad (1.5)$$

1.9.4 RANDOM FOREST

Random forest is widely used for classification and regression models. It is a supervised Machine Learning algorithm which builds multiple decision trees. In case of classification, it takes a majority vote but in case of regression it takes an average. It has better performance in classifications problems, Random Forest uses.

CHAPTER 2

LITERATURE REVIEW

DoH stands for DNS over HTTPs, it is a very novel technology, it was introduced in 2018 and was published as RFC 8484.[6] It was introduced in order to improve the security of the network and the privacy or the users. Unlike the DNS, where the queries were sent as plain text, it performs the DNS resolution over HTTPs. It is continuously evolving since its introduction; it is still a work in progress.

Many browsers have started providing this feature already. In 2018 itself Google and Firefox started their testing. And by 2020 Firefox has switched to DOH by default in the United States. But it is yet to pick up the pace. Even though but not much have been presented in the field but some papers are there which have already been published.[8]

Paper by Borgolte et al. [6]presented a white paper in which they talk about DoH in general. This paper provides a discussion on various aspects of doh such as performance, privacy and security.[9] But there is no network level analysis presented in this paper. They have considered the internet space as marketplace. they have then tried analyzing the possible effects on the various entities present in the marketplace including the consumers. They have also compared the performance of Doh and plain DNS

Hjelm et al. [10]of SANS institute presented a paper on the description of Doh services. They have used RITA framework. RITA stands for Real time Intelligence Analytics. They tried analyzing the logs in order to analyze the network traffic. They have provided with the ways to prevent or block the DoH on the organization network

Patsakis et al.[11] in their paper has emphasized on botnet and DGA. These botnets used DNS as communication channels for establishing a communication between the command servers and the control servers. They have also talked about how DAGs are attacked to generate large number of fake domains which makes providing security even more challenging.

Bushart et al and Siby et al. worked on Alexa in order to identify the encrypted traffic. They have used messages in busts and sequences. They have also talked about how encrypted DNS can still be analyzed at traffic level making it sensitive to monitoring. To overcome this problem, it was suggested to introduce the padding before encryption. It was proposed by RFC 8467. They have majorly focused on identifying the fingerprinting the said websites.

P. Hoffman in DNS queries over https has proposed a standard and has defined the protocol. It explains how the DNS queries can be sent over https and how to get their response for https. It also talks about how each and every query is mapped in this exchange.

Dmitrii Vekshin[12] in their paper have worked on the encrypted network analysis. They have created the DOH dataset and made it available to the public. In their paper they have implemented machine learning on the dataset to in order to categorise the traffic as DoH or Non DoH and further classify it as malicious or benign. In their paper they have used 5 difference algorithms for classification. They have used KNN,

Random Forest, C4.5 Tree, Ada Boost, Naïve Bayes algorithms. They have been able to get a good accuracy of 99%.Their approach is different from what has been implemented my this project.

In this project we have used Artificial Neural Network for the classification of DoH. Our model has outperformed all their previous work. Our model was able to predict everything accurately with just 2 false positive. The overall accuracy of the model was 100% .

CHAPTER 3

METHODOLOGY

3.1 DATASET

CIRA-CIC-DoHBrw[13] is 2020 data set which was publicly made available by the Canadian Institute for Cybersecurity. This dataset was manually created and was later released for the public. It was created by capturing the traffic from browsers which had doh enabled. Thousands of top websites were visited and with the help of DNS and tunneling this data was captured. The data collected was in PCAP which were converted to prevent user privacy. These were later converted using the 'flow_meter'.

This dataset is divided into 2 levels consisting of a total of 4 CSV files. On the first level data classification was done to segregate it into DoH and Non-Doh traffic. The first level has 2 CSV files and it distinguishes between the doh and non doh data. On the second level time series classification. Was carried out to segregate the traffic as malicious or benign. second level again has 2 CSV files and it distinguishes between malicious and non-malicious doh.

Once the data was captured its dimensionality was reduced in order to skip the unnecessary information which was either too small or insignificant. This was done using the concept of packet clumps. When a flow of consecutive packets is sent which has the same source and same destination then it is called as packet clumps. Packet clamping is done in order to identify the packets which are the part of same process but are separated due to fragmentation.

3.1.1 CATEGORIZATION OF DATA IN THE ORIGINAL DATA SET

- Non-Doh: Traffic generated by a browser which is using plain DNS and HTTPS is classified as non-DoH

- DoH: Traffic generated by a browser which has DoH enabled and is using HTTPS is classified as DoH traffic.

- Benign-DoH : It is the DoH traffic which was classified as non- malicious

- Malicious-DoH: This is the malicious traffic which was generated with the help of tunnelling tools. Few examples of these are DNSCat2, Iodine. These sent malicious TCP traffic in DNS searches.

The original dataset has 2,69,643 rows and 35 features. Those features are depicted in the table below

SourceIP
DestinationIP
SourcePort
DestinationPort
TimeStamp
Duration
FlowBytesSent
FlowSentRate
FlowBytesReceived
FlowReceivedRate
PacketLengthVariance
PacketLengthStandardDeviation
PacketLengthMean
PacketLengthMedian
PacketLengthMode
PacketLengthSkewFromMedian
PacketLengthSkewFromMode
PacketLengthCoefficientofVariation
PacketTimeVariance

PacketTimeStandardDeviation
PacketTimeMean
PacketTimeMedian
PacketTimeMode
PacketTimeSkewFromMedian
PacketTimeSkewFromMode
PacketTimeCoefficientofVariation
ResponseTimeTimeVariance
ResponseTimeTimeStandardDeviation
ResponseTimeTimeMean
ResponseTimeTimeMedian
ResponseTimeTimeMode
ResponseTimeTimeSkewFromMedian
ResponseTimeTimeSkewFromMode
ResponseTimeTimeCoefficientofVariation
Label

Table 3.1 Features of the dataset

3.2 MACHINE LEARNING MODEL USED

3.2.1 ARTIFICIAL NEURAL NETWORK

In human brain and nervous system neural are fundamental units, they are responsible for carrying all the sensory inputs to the brain and the response from the brain. There are large number of neurons in brain and they for a network. Similarly Artificial neural network is an attempt to replicate the idea of human brains using algorithms. Structure of biological neuron is described in the figure.

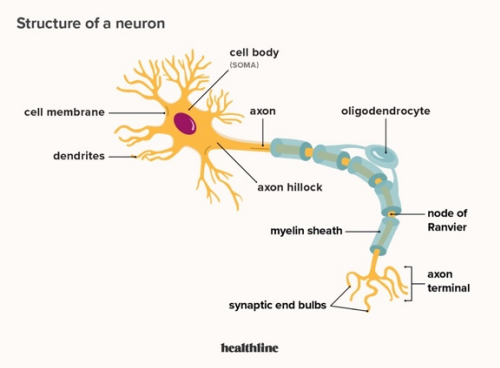

Fig 3.1 Biological Neuron[14]

Even though the idea is to copy the human brain still there are some differences between the two. Neurons present in human body carry out parallel processing whereas artificial neurons carry out sequential processing.

Despite their differences an artificial neuron can be compared with biological neuron. It has been described in the table

Biological Neuron	ANN
Dendrites	Inputs
Cell Nucleus	Nodes
Synapse	Weights
Axon	Output

Table3.2 Comparison between ANN and biological Neuron

3.2.2 ARCHITECTURE OF ANN

For mimicking the human brains we need to have large numbers of neurons. These neurons are arranged in multiple layers, there is a sequence between these layers.

The most basic form of ANN consists of three layers.

- Input layer
- Hidden Layer
- Output layer

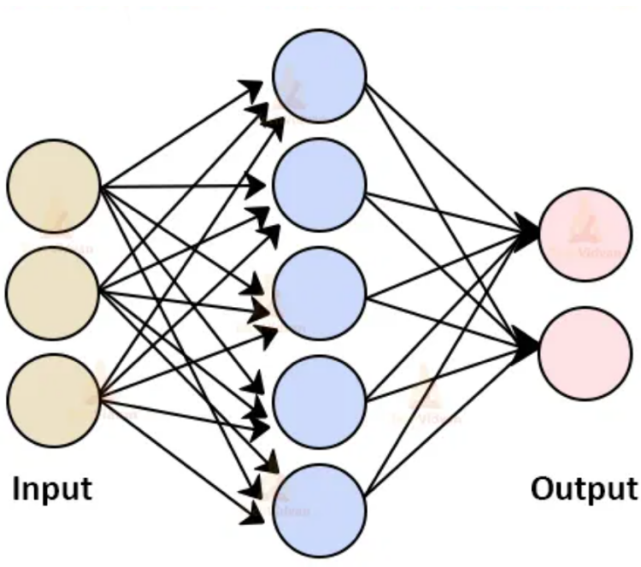

Fig 3.2 Artificial Neural Network

Input Layer
This layer is responsible to accepting the inputs passed to it

Hidden layers

This Is the layer in been the input and the output layer. All the calculations are performed in these layers. These are done in order to find the features and patterns present in the dataset.

Output layer
Hidden layer, after transforming the input layer passes the result to the output layer

3.2.3 PERCEPTRON

In 1957, Frank Rosenblatt introduced the perceptron. It was used in supervised learning for binary classification.

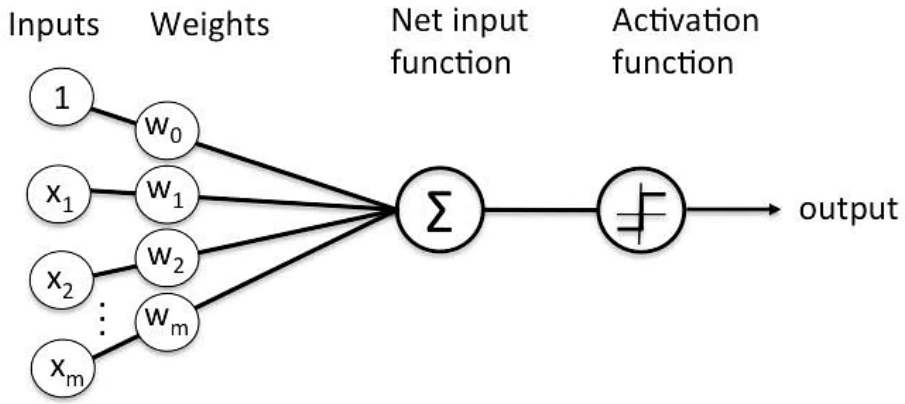

Fig 3.3 Perceptron

Types of perceptron

Single layer

These are only fir for learning patterns which can be linearly separated. The weight is calculated by adding up the input weights, based on the value they are activated. If the result is not satisfying the changes are to be made to the input weights.

Multi-layer

These have at least two hidden layers and these are feedforward network. The operation takes place in two steps; forward step and the backward step. In the former the activation value moves from input to output layer. In case of later case demanded value moves back from the output for modification.

Formula

$$\sum_{i=1}^{m} Wi * Xi \quad (3.1)$$

$$Wn = Wn - \eta * \frac{\partial J}{\partial Wn} \quad (3.2)$$

3.3 DATA PREPROCESSING

Data preprocessing is carried out in order to make the dataset suitable for computations. Null values are either removed or calculated using suitable methods

This dataset had 344 missing values in "**ResponseTimeTimeMedian**" field and 344 missing values in "**ResponseTimeTimeSkewFromMedian**". Since is data set is very large so dropping those rows wont affect the model. Therefore those values were dropped

```
df = df.dropna().reset_index(drop=True)
```

Fig 3.4 code for dropping the null values

The "**Timestamp**" values were in raw format and could not be used directly for the computation. It was first converted into seconds. Since the dataset is time series therefor timestamp played a very important role in uplifting the performance

```
df['TimeStamp'] = pd.to_datetime(df['TimeStamp'])       # Convertin

def timeToSeconds(data):        # Function to convert time in secon
    return data.hour*60*60 + data.minute*60 + data.second

df['TimeStamp'] = df['TimeStamp'].map(timeToSeconds)  # converting
```

Fig 3.5 Code for timestamp conversion

The "**sourceIP**" and "**destinationIP**" also needed to be changed. This was done by using one hot encoder and generating dummy columns

```
# One hot encoding of SourceIP and DestinationIP as both are categorical
for col in ['SourceIP', 'DestinationIP']:
    df = pd.merge(left=df,
                  right=pd.get_dummies(df[col], prefix=col, prefix_sep='_'),
                  left_index=True,
                  right_index=True
    )
    df = df.drop(columns = [col])
```

Fig 3.6 Code for changing source and destination IP

In the alternate approach this model was implemented by splitting SourceIP and DestinationIP into 4 Separate columns and then applying the one hot encoder on the newly created 8 columns. After this the 2 original columns were dropped.

```
cat_cols = ["DestinationIP1", "DestinationIP2","DestinationIP3","DestinationIP4"]
cat_cols_encoded = []
for col in cat_cols:
  cat_cols_encoded += [f"{col[0]}_{cat}" for cat in list(df2[col].unique())]

cat_cols_encoded


oh_encoder = OneHotEncoder(sparse=False, handle_unknown='ignore')
encoded_cols = oh_encoder.fit_transform(df2[cat_cols])
df_enc = pd.DataFrame(encoded_cols, columns=cat_cols_encoded)

df_oh = df2.join(df_enc)
df_oh
```

Fig 3.7 Code for splitting the columns

Lastly the label had to be changed into binary.

```
df = df.replace({'Label':{'Benign':0,'Malicious':1}})
```

Fig 3.8 Code for changing the label

3.4 FEATURE SCALING

It is one of the most important task while carrying put pre processing

As there is large difference in the feature values, they must be converted to a common scale. In this project this was done using the max min scalar. It scales the data in the range of 0 and 1

```
scaler = MinMaxScaler(feature_range=(0, 1))
df[col_trans] = scaler.fit_transform(df[col_trans])
```

Fig 3.9 Code for feature scaling

3.5FEATURE SELECTION

In this step the dimensionality needed to be reduced. Our dataset had 71 columns are preprocessing.

Principal Component Analysis (PCA) was used for carrying out the feature selection. The major advantage was that it removed all the correlated features. After carrying out this step 31 features captured the 95% variance of the data. Thus, we transformed the original data into 31 principal components

```
pca = PCA().fit(X)
explained_var = np.cumsum(pca.explained_variance_ratio_)
index_95 = np.where(np.array(explained_var) >= 0.95)[0][0]+1
```

Fig 3.10 Code for PCA

3.6 IMPLEMENTATION

This model is implemented using Neural Network. It has been implemented in Keras. The model has 1 input layer, 5 hidden layers, and 1 output layer. It has been depicted in the figure

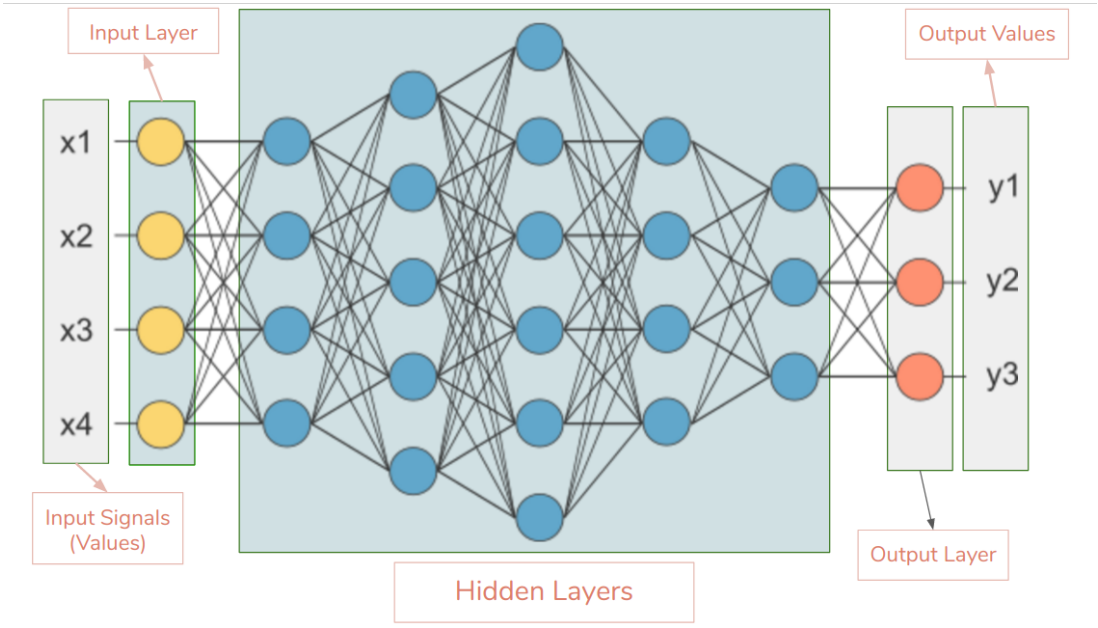

Figure 3.11 Number of layers in the model

The input layer had 132 neurons. There was a total of 5 hidden layers. First hidden layer had 64 neurons, second hidden layer had 32 neurons, third hidden layer had 16 neurons, the fourth layer had 8, fifth layer had 4. The output layer had 2 neurons

RELU was used as the activation function in the input and hidden layers. Softmax was used in the output layer. The model was initialized with Adam optimizer with a learning rate of 0.001

CHAPTER 4

RESULT

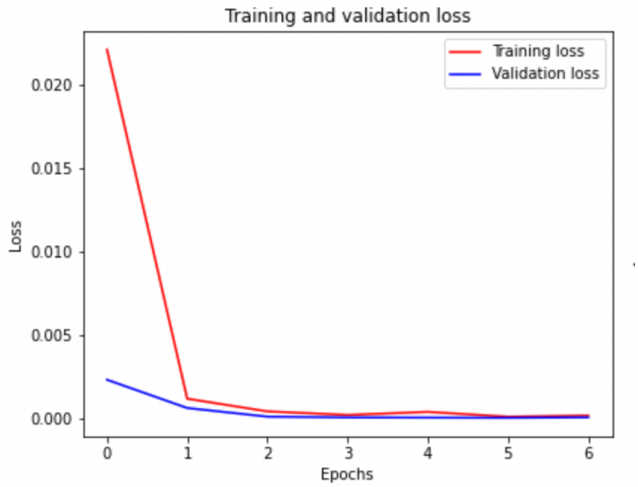

Fig 4.1 Training and validation loss

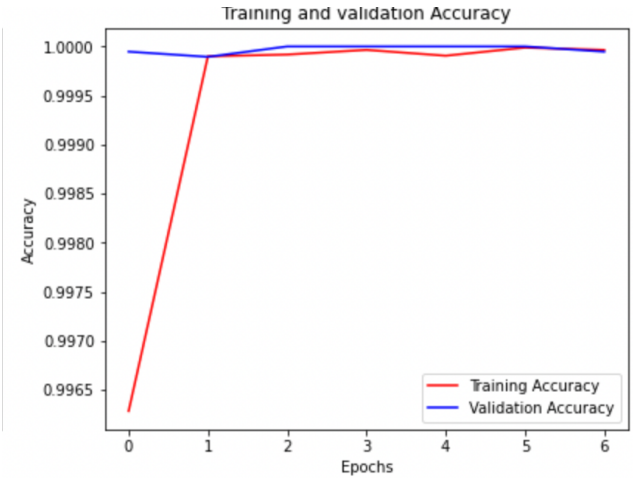

Fig 4.2 Training and validation accuracy

In the above figure we can clearly see that both the training loss and validation loss are equal and are minimized. Similarly, the training accuracy and validation accuracy are same and they are maximined. Therefore, we can safely conclude that the model has been trained well and there is no over fitting of the training data.

The model was able to predict the results with great accuracy in both training and testing data.

In the first approach where “sourceIP” and “destinationIP” was converted using one hot encoder and generating dummy columns, the model showed exceptional accuracy. In the training data it predicted only two values as false positive whereas in case of testing data it predicted no wrong output.

In this approach the model got 100%

```
                Train output:

Accuracy:  1.0

True Positive Rate: 1.0
False Positive Rate: 0.0

Classification Report:
               precision    recall  f1-score   support

           0       1.00      1.00      1.00     13739
           1       1.00      1.00      1.00    174770

    accuracy                           1.00    188509
   macro avg       1.00      1.00      1.00    188509
weighted avg       1.00      1.00      1.00    188509
```

Fig 4.3 Precision, Recall and Accuracy in train set in first approach

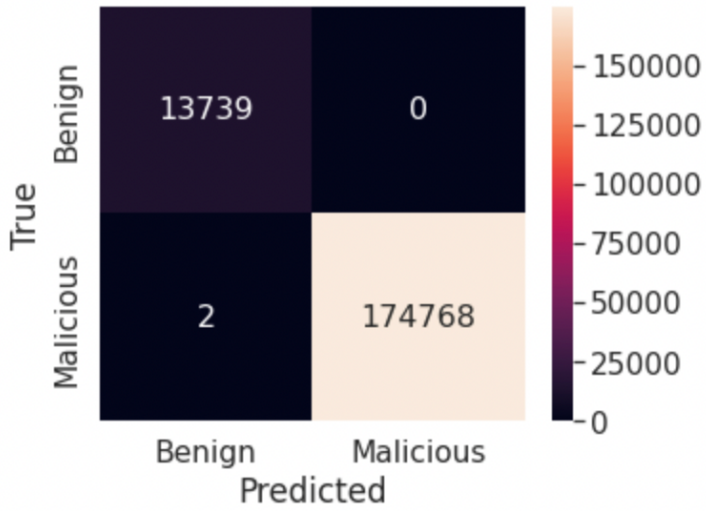

Fig 4.4 showing Confusion matrix in the train set in first approach

```
                Test output:

Accuracy:  1.0

True Positive Rate: 1.0
False Positive Rate: 0.0

Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      6007
           1       1.00      1.00      1.00     74783

    accuracy                           1.00     80790
   macro avg       1.00      1.00      1.00     80790
weighted avg       1.00      1.00      1.00     80790
```

Fig 4.5 Precision, Recall and Accuracy in test set in first approach

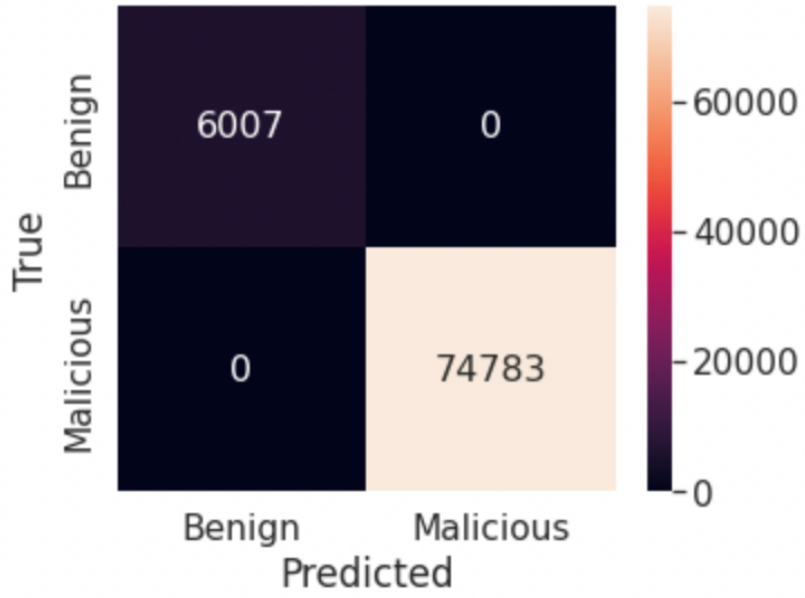

Fig 4.6 Confusion matrix in test set in first approach

In the second approach where the “SourceIp” and “DestonationIP” were first split into 4 separate columns , the model predicted 180 wrong values in the training set and 73 wrong values in the testing set. **Despite this the model achieved an F! score of 99% and overall accuracy of more than 99%**

```
                Train output:

Accuracy:  1.0

True Positive Rate: 1.0
False Positive Rate: 0.01

Classification Report:
               precision    recall  f1-score   support

           0       1.00      0.99      0.99     13739
           1       1.00      1.00      1.00    174770

    accuracy                           1.00    188509
   macro avg       1.00      0.99      1.00    188509
weighted avg       1.00      1.00      1.00    188509
```

Fig 4.7 Precision, Recall and Accuracy in train set in second approach

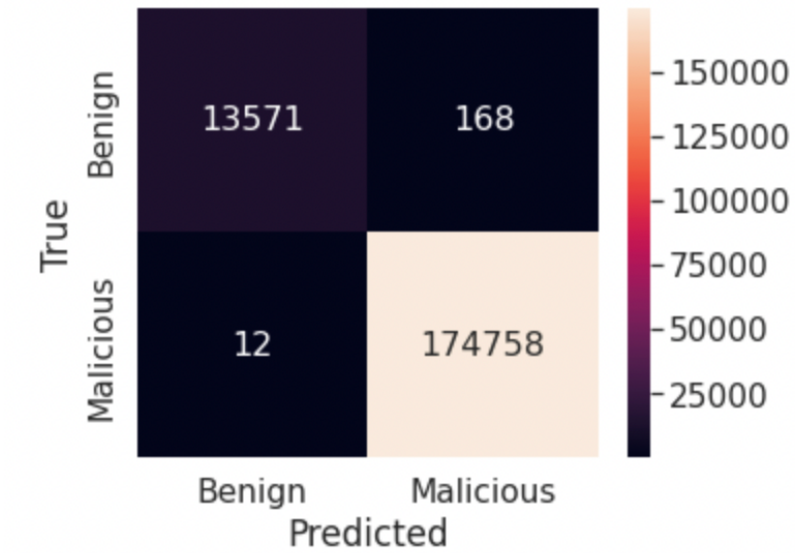

Fig 4.8 Confusion matrix in train set in second approach

```
                Test output:

Accuracy:  1.0

True Positive Rate: 1.0
False Positive Rate: 0.01

Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.99      0.99      6007
           1       1.00      1.00      1.00     74783

    accuracy                           1.00     80790
   macro avg       1.00      0.99      1.00     80790
weighted avg       1.00      1.00      1.00     80790
```

Fig 4.9 Precision, Recall and Accuracy in test set in second approach

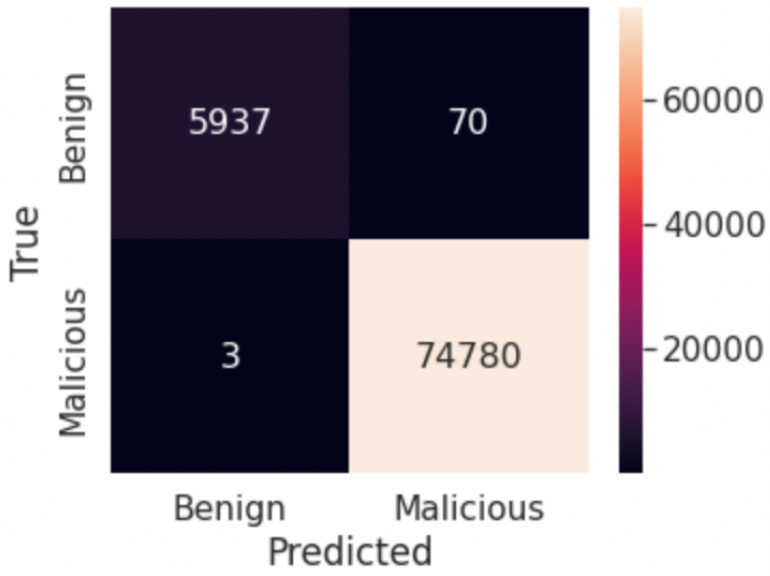

Fig 4.10. Confusion matrix of test set in second approach

CHAPTER 5

CONCLUSION

As users are getting more and more concerned about their privacy in the online world the use of DoH is certain to pick up the pace. This will get an additional boost from the race between the companies to control the major portion of doh traffic. In such a situation it will open back doors for other problems. With the help of this project, we have tried to show how we can successfully use machine learning algorithm so accurately classify the DoH traffic as benign or malicious.

5.1 FUTURE SCORE

In the future the implementation can be done at the IDS level in order to block certain packets from entering into the network. This can easily be done once they have been classified as malicious.

As the Doh protocol is evolving so there is a need for more datasets and information gathering. DNS can also be replaced by DoT in future; therefore, models can be made more generalized so as to include TSL connections instead of focusing on one aspect of it.

REFERENCES

[1] K. Borgolte *et al.*, “How DNS over HTTPS is Reshaping Privacy, Performance, and Policy in the Internet Ecosystem.” [Online]. Available: https://ssrn.com/abstract=3427563

[2] H. Serhii and P. Cynthia, “Advantages of DNS-over-HTTPS over DNS,” 2020.

[3] K. Bumanglag and H. Kettani, “On the impact of DNS over HTTPS paradigm on cyber systems,” in *Proceedings - 3rd International Conference on Information and Computer Technologies, ICICT 2020*, Mar. 2020, pp. 494–499. doi: 10.1109/ICICT50521.2020.00085.

[4] C. Patsakis, F. Casino, and V. Katos, “Encrypted and covert DNS queries for botnets: Challenges and countermeasures,” *Computers and Security*, vol. 88, Jan. 2020, doi: 10.1016/j.cose.2019.101614.

[5] D. Vekshin, K. Hynek, and T. Cejka, “DoH Insight: Detecting DNS over HTTPS by machine learning,” Aug. 2020. doi: 10.1145/3407023.3409192.

[6] T. Böttger *et al.*, “An empirical study of the cost of DNS-over-HTTPS,” in *Proceedings of the ACM SIGCOMM Internet Measurement Conference, IMC*, Oct. 2019, pp. 15–21. doi: 10.1145/3355369.3355575.

[7] P. Soucy and G. W. Mineau, “A Simple K" Algorithm for Text Categorization.”

[8] H. Serhii and P. Cynthia, “Advantages of DNS-over-HTTPS over DNS,” 2020.

[9] K. Borgolte *et al.*, “How DNS over HTTPS is Reshaping Privacy, Performance, and Policy in the Internet Ecosystem.” [Online]. Available: https://ssrn.com/abstract=3427563

[10] "Global Information Assurance Certification Paper A New Needle and Haystack: Detecting DNS over HTTPS Usage GIAC (GCIA) Gold Certification A New Needle and Haystack: Detecting DNS over HTTPS Usage 2 Drew Hjelm, drew@vets.io," 2019. [Online]. Available: http://www.giac.org/registration/gcia

[11] C. Patsakis, F. Casino, and V. Katos, "Encrypted and covert DNS queries for botnets: Challenges and countermeasures," *Computers and Security*, vol. 88, Jan. 2020, doi: 10.1016/j.cose.2019.101614.

[12] D. Vekshin, K. Hynek, and T. Cejka, "DoH Insight: Detecting DNS over HTTPS by machine learning," Aug. 2020. doi: 10.1145/3407023.3409192.

[13] DoH Dataset, "Dataset."

[14] Biological Neuron, "Biological Neuron."

PAPER NAME

Saurav_Kumar_Thesis.pdf

AUTHOR

Saurav Final

WORD COUNT

5691 Words

CHARACTER COUNT

27962 Characters

PAGE COUNT

43 Pages

FILE SIZE

4.3MB

SUBMISSION DATE

May 26, 2022 7:13 AM GMT+5:30

REPORT DATE

May 26, 2022 7:14 AM GMT+5:30

11% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 9% Internet database
- 2% Publications database
- Crossref database
- Crossref Posted Content database
- 10% Submitted Works database

Excluded from Similarity Report

- Bibliographic material
- Quoted material
- Cited material
- Small Matches (Less then 8 words)
- Manually excluded sources

11% Overall Similarity

Top sources found in the following databases:

- 9% Internet database
- Crossref database
- 10% Submitted Works database
- 2% Publications database
- Crossref Posted Content database

TOP SOURCES

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	**dspace.dtu.ac.in:8080** Internet	4%
2	**61.16.156.122** Internet	2%
3	**wbdg.org** Internet	1%
4	**repository.seku.ac.ke** Internet	<1%
5	**University of Nottingham on 2009-01-19** Submitted works	<1%
6	**pergamos.lib.uoa.gr** Internet	<1%
7	**irrigation.org** Internet	<1%
8	**mdpi.com** Internet	<1%

9	**Universiti Teknologi MARA on 2015-01-20** Submitted works	<1%
10	**University of Northampton on 2022-05-14** Submitted works	<1%
11	**Norman Dziengel, Martin Seiffert, Marco Ziegert, Stephan Adler, Stefan...** Crossref	<1%
12	**mafiadoc.com** Internet	<1%
13	**Heriot-Watt University on 2020-07-24** Submitted works	<1%
14	**Queen's University of Belfast on 2021-12-15** Submitted works	<1%
15	**University of Macau on 2019-05-21** Submitted works	<1%
16	**ebin.pub** Internet	<1%
17	**gis.inf.elte.hu** Internet	<1%
18	**Arrigo Palumbo, Vera Gramigna, Barbara Calabrese, Nicola Ielpo. "Mot...** Crossref posted content	<1%
19	**Heart of Worcestershire College on 2020-02-05** Submitted works	<1%
20	**Liverpool John Moores University on 2022-03-08** Submitted works	<1%

21	Rafa Alenezi, Simone A. Ludwig. "Classifying DNS Tunneling Tools For ... Crossref	<1%
22	University College London on 2017-05-30 Submitted works	<1%
23	University of Western Sydney on 2018-09-25 Submitted works	<1%
24	Yueyang Liu, Zhinoos Razavi Hesabi, Mark Cook, Levin Kuhlmann. "Epil... Crossref	<1%
25	arxiv.org Internet	<1%
26	citeseerx.ist.psu.edu Internet	<1%

Excluded from Similarity Report

- Bibliographic material
- Cited material
- Manually excluded sources
- Quoted material
- Small Matches (Less then 8 words)

EXCLUDED SOURCES

Delhi Technological University on 2019-06-27 4%

Submitted works