Feed Forward MLP SPAM domain Detection Using Authoritative DNS Records and Email Log

MSc Internship
Msc. Cybersecurity

Chirag Sharma
Student ID: x18151485

School of Computing
National College of Ireland

Supervisor: Dr. Muhammad Iqbal

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Chirag Sharma		
Student ID:	X18151485		
Programme:	MSc. Cybersecurity	**Year:**	2019-2020
Module:	Cybersecurity		
Supervisor:	Dr. Muhammad Iqbal		
Submission Due Date:	8 Jan 2020		
Project Title:	Feed Forward MLP SPAM domain Detection Using Authoritative DNS Records and Email Log		
Word Count:	6772 word		
Page Count	17		

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.
I agree to an electronic copy of my thesis being made publicly available on NORMA the National College of Ireland's Institutional Repository for consultation.

Signature:

Date: 8 Jan 2020

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□

You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Feed Forward MLP SPAM domain Detection Using Authoritative DNS Records and Email Log

Chirag Sharma
X18151485

Abstract

Email has been the main source of business communication. Online data loss prevention vendors estimate about 15% of global email space contributes to spam. With emergence of new and highly adaptive spams, attackers leverage the botnets creating large IP address pool which can be used for domain spoofing and prevent domain takedown. There has been development of new technologies such as SPF and DKIM to provide sender authenticity and integrity, but they are not sufficient themselves for spam filtering and the policies related to SPF and DKIM cannot be harsh because these technologies have not been implemented fully by each of reputable domains hence it could lead to most of the legitimate emails undelivered. To solve this problem, this paper proposes the mechanism of detection of spam domains using machine learning with use of Email characteristics such as DKIM signature domain, List-Unsubscribe feature and active DNS records such as SPF record, Authoritative nameservers etc. We have been able to achieve 97.11% accuracy by applying Feed Forward Neural Network machine learning model and with accuracy of 95% which is more than the previous study by taking into consideration List-unsubscribe feature which is actively used by spam domains.

Index-terms – Spam Detection, Active DNS records, Feed Forward neural Network

1. Introduction

Phishing involves some tricky emails sent from either spoofed reputable domains or sent from a botnet in a bulky and unsolicited way primarily for identity theft or in some cases used for deploying malware on to victim's system. This problem persists for over decades and it's still very effective crest for exploitation and has caused losses of multi-billions to almost all sectors of global market.

If we take an example of Fortune 500 company the main priority of the Information Security team of that company is to have strong data loss prevention mechanism. In today's world there are lot of proprietary solutions for data loss prevention deployed on the outward facing mail exchange servers but most of them focus on regex/pattern matching against the text of the email such as keywords *Bitcoin*, *Urgent Attention* or matching the domain name against the DNSBLs. Sometimes DNSBLs are not up to date and with highly adaptive nature of spamming systems such as fast flux networks in command and control of bot master, these types of emails can be easily bypassed.

An extensive research has been going on to develop strong and robust solutions to cater to domain spoofing done by botnets. There has been development of authentication mechanisms like DKIM (Domain Key Identification Mail) **[1]** and SPF (Sender Policy Framework) **[2]** which can be added in DNS records which can verify a legitimate domain. But this is restricted to verification DNS records. It does not cater to the fact that spammers could send emails from compromised legitimate email addresses also. Another approach to solving the problem is domain blacklisting but it is really hard to curate those lists especially when spammer use botnets and create fast flux networks which consists of a lot of IP addresses affiliated with multiple second-level domain name. This is because the spam-campaigns are run through botnet, various phishing emails with slight changes in headers or the body are generated from the same botnet. Hence this becomes difficult to maintain a blacklist because this can be easily bypassed. Hence, there is a need of certain confrontation mechanism which analyze both the aspects of phishing such as email headers which consist as well as DNS records of Email domain.

This paper proposes the implementation of such mechanism to detect phishing domains with high accuracy using email reception log from a honeypot **[3]** and we use active DNS measurements data from OpenIntel **[4]** which is an opensource platform that collects responses from queries sent to authoritative DNS servers. We create a dataset using the above-mentioned sources and apply machine learning algorithms to build a classifier. The reason to use active DNS measurements rather than DNS cache server is that cache servers are located locally with the ISP's and the data might be outdated and because they are recursive in nature most of the results are cached hence the TTL(Time-To-Live) values might not be reliable. Another major reason for not choosing the cached servers because of cache poisoning in which perpetrators update the cached zone file with IP address of malicious domains, hence the user lands on to that malicious site rather than the legitimate site for which he has requested.

There has been a similar method proposed in **[5]**, the accuracy is not that high.

2. Related Work

2.1 DNS Analysis

There are various methods which use passive DNS data obtained from cache servers has been used to build a domain reputation system in **[6]** which assigns scores to domains associated with spam or phishing. The features used in this work were BGP (Border gateway Protocol), Autonomous systems Number (ASN) assigned to multiple subdomains which have proven to play a significant role in identifying botnets. But the drawback of this research was that it cannot generate scores for newly created domains because passive DNS servers do not have information because these domains have not been cached.

Another research using active DNS data has been done in **[7]** which is also a domain reputation mechanism which does reconnaissance at the time domain is registered, which helps in detection of malicious domains in the early stages, which can help network operators to update their email gateways with a pre-emptive list. This research uses Convex Polytype Machine which is a supervised Machine Learning algorithm with features including date:time of registration/length of registration etc. But this work focuses on domain registered under single zone. So, the spammers can register domains under different zones and TLDs and nowadays there is a rise in expansion of different TLDs apart from some generic ones.

To avoid DNSBLs (DNS Blacklists) spammers use different host names within a second level domain name or using third level domains as a result creating randomness and large pool of spam domain URI. Through fast flux networks an attacker can provide its IP address in the DNS server of compromised system as a result when sender queries a legitimate domain, it gets the webpage of the attacker containing malwares. The research proposed by **[8]** analyzes number of A-records for a fully qualified domain name fetched from the email address and TTL values of A record and found that 89% of the spam domains had high TTL value i.e. between 1801-3600 which shows that spammer keep their DNS records cached for shorter time to achieve higher availability and fast flux networks get an opportunity to abuse DNS round robin.

Passive DNS analysis can also give a lot of information fast flux networks because the traffic available at disposal somehow contains huge number of requests from a compromised host which is under command and control of bot master which is the owner of malicious content. One such study **[6]** uses data from recursive DNS servers using time-based features such as TTL(Time-to-live) of each query of a record made. Number of IP addresses, country code, reverse_dns query results etc. This technique showed higher accuracy in detecting malicious domains, but this technique only works if the network is unmonitored and traffic is not abruptly changing.

2.2 Spam Identification

The techniques generally used for detecting phishing emails in known researches generally involve analyzing text of the email through means of natural language processing. Techniques such as Keras word embedding has been used in **[9]** modelled with Tensor Flow. It also adds one extra pooling layer which reduces redundant dimensions, hence generating a better result. The model chosen was able to develop accuracy of 94.2%. The dataset used was bifurcated into two parts one with email headers and one without email headers.

Headers of an email tells a lot about from where that particular email has come from. It's like a normal letter which contains multiple envelopes stating the proof of origin as well as intermediate mail junctions. Email header contain useful features which can be used to identify domain spoofing. **[10]** makes use of email headers to detect spams by using features such as address validation, Number of relay servers in between, Message ID domain match etc. and detects phishing domains using multiple algorithms and then compared the result with each other getting highest accuracy of 99.2%.

Research based on the origin of the spam email i.e. the domain registration has not been conducted so extensively. Yet **[11]** conducts a survey of such approaches and classified them based on the factors like Type of data collection(active/passive), Data Enrichment i.e. more and more relevant features along with DNS such as WHOIS lookup, GEOIP lookup etc. which can help detection of such mails significantly. The paper also chooses Algorithm design and Evaluation methodology of each of the research.

Both supervised as well as semi supervised approach and each of the approaches come different set of challenges. For example: Ground truth of malicious domain which are posted on blacklists are labeled as same, a malicious domain can be spam, phishing, adult content etc. But mostly on the open source blacklists it is labeled as 'SPAM'. Another challenge faced is the inclusion of FQDN (Fully Qualified domain name) or 2nd level domain name because each of the subdomains might have different reputation that its 2nd level domain.

Right feature selection right source is also a bigger challenge because most of the DNS data available is through passively recursive resolvers which is not accurate and persistent and it is very hard to fetch Active DNS data because the servers are authoritative and most of the time does not reply exactly with the queries. Choosing the right detection method is also very important because adversaries use highly adaptive approaches to bypass the detection systems or gateways and the models have to be reconfigured in order to capture those new changes. Adding to the fact a spam campaign can just last for few hours before it completely vanishes out from the ecosystem.

2.3 Detection Method

Although machine learning is the best way for predictive analysis in case the data is already labeled. **[12]** has used unsupervised method for classification and works on packet level traffic capture. Work in **[6]** uses decision trees and clustering for analysis of passive DNS records. **[7]** also uses clustering techniques based on domain and zone features. **[13]** uses SVM model against a private dataset of active DNS as well as e-mail reception log. Most of the techniques used show decent accuracy and precision but the problem with those techniques they do not involve backpropagation and analysis of the hidden layers. **[14]**

3. Research Methodology

In this section the methodology for this particular research is discussed in aspects of Business Understanding, Data Recognition, Data Preparation as we follow CRISP-DM as a standard which is widely known across the industry benchmarks. The reason for choosing CRISP-DM methodology is that it is business oriented which means the solution, we provide can be used in information security industry.

3.1 Business Objective

There has been an introduction of some new and emerging technologies such as SPF and DKIM in DNS records, to tighten the email security. Sender verification is the first step to email data loss prevention and SPF and DKIM ensure sender's legitimacy. But these two are unfortunately not the spam filters. Because of advancements in DNS as well availability of more TLDs, almost all the spam domains have SPF and DKIM records but the values inside those records are from legitimate but compromised hosts hence spammers are able to hide their true identity. The businesses also cannot put zero tolerance policies for SPF and DKIM policy checks as most of the legitimate emails will also bounce back along with spam ones. Hence, we need data of emails to analyze so we can understand SPF, DKIM records of malicious domains and other DNS characteristics of emails and their sender domains once email has been passed by the recipient mail server and can predict the reputation of those domains based on previously classified examples.

But due to data protection rules and regulations organizations cannot share the email logs for research outside. The only option remaining is to set up a honeypot and create a dataset but the problem with this data collected is that spammers do a bit of social engineering about the users and their email addresses which makes this research even more complex and the dataset more imbalanced. Coming onto DNS

3.2 Data Understanding and Collection

For the purpose of this detection we need a data containing large number of e-mails containing both malicious and benign ones. **[3]** has hosted one such archive which is a honeypot collecting mails with each mail in the form of text file which is raw and contains all the headers as it is. From the headers we extracted 'Received-From' and extracted domain name from it and then we extracted 'DKIM-Signature' and 'List-Unsubscribe' to match whether the domain name matches the corresponding DKIM Signature domain as well as in the List Unsubscribe. As per the study from **[15]** the list unsubscribe link can be malicious and can share lot of information about the victim to the attacker, hence we consider this as an important feature to use in our model. We match the domain specified in DKIM and List-unsubscribe if it is a match, we mark the corresponding cell as yes, for an un-match we mark it as no and if the record is not present, we mark it as the same.

After extraction of domain we query the DNSBL to get the reputation as a feature. Then we query the query the OpenINTEL database which is the largest repository for authoritative DNS data and currently measures about 60% of the global domain name space. **[5]**

We take the 'A_record' and use 'response_ttl' of the record as a feature, similar is done with 'NS', 'MX' and 'TXT' records. The TTL values are in terms of seconds. We choose TTL as a feature because most of the spam domains have a low TTL because once the mail is sent from malicious domain the DNS resolver of the victim's network will cache it for short duration of time and after a certain point of time the record will be vanished from the server, hence it will be difficult to trace it because of the round robin technique used, the IP address of the spam domain will be marked as redundant by that DNS server because of short TTL **[16]**. **[6]** has validated that TTL in malicious domains changes rapidly and can have multiple TTL values showing at same time.

We also chose number of IPs associated with A record, number of nameservers associated with NS record and number domain servers associated with MX record. We also chose 'TXT' record values in a way to check whether SPF records consists of 'include' statement record or not. SPF contains a range values which depict the type of authorization policy ('-all', '~all', '+all', ?all). A reputable domain would not allow any mail which is not interest especially from a spammer so they will put '-all' policy which does not allow any mail if IP mentioned in TXT record or IP of MX domains does not match with the SenderIP. Most of the spammers will either would not have SPF records or to seem legitimate to a normal user they will have '~all' or '-all'.

Hence, we have chosen 'Type of All' as a feature. Table 1 shows the aggregation of all the features chosen. Unfortunately, due to huge domain space to look for active DNS records and number active spams available are very less we were able to fetch only 802 rows of data for this research.

Feature	**Extracted From**	**Type**
Country_code	DNS Record	Categorical
response_ttl_A	DNS Record	Numerical
response_ttl_MX	DNS Record	Numerical
response_ttl_NS	DNS Record	Numerical

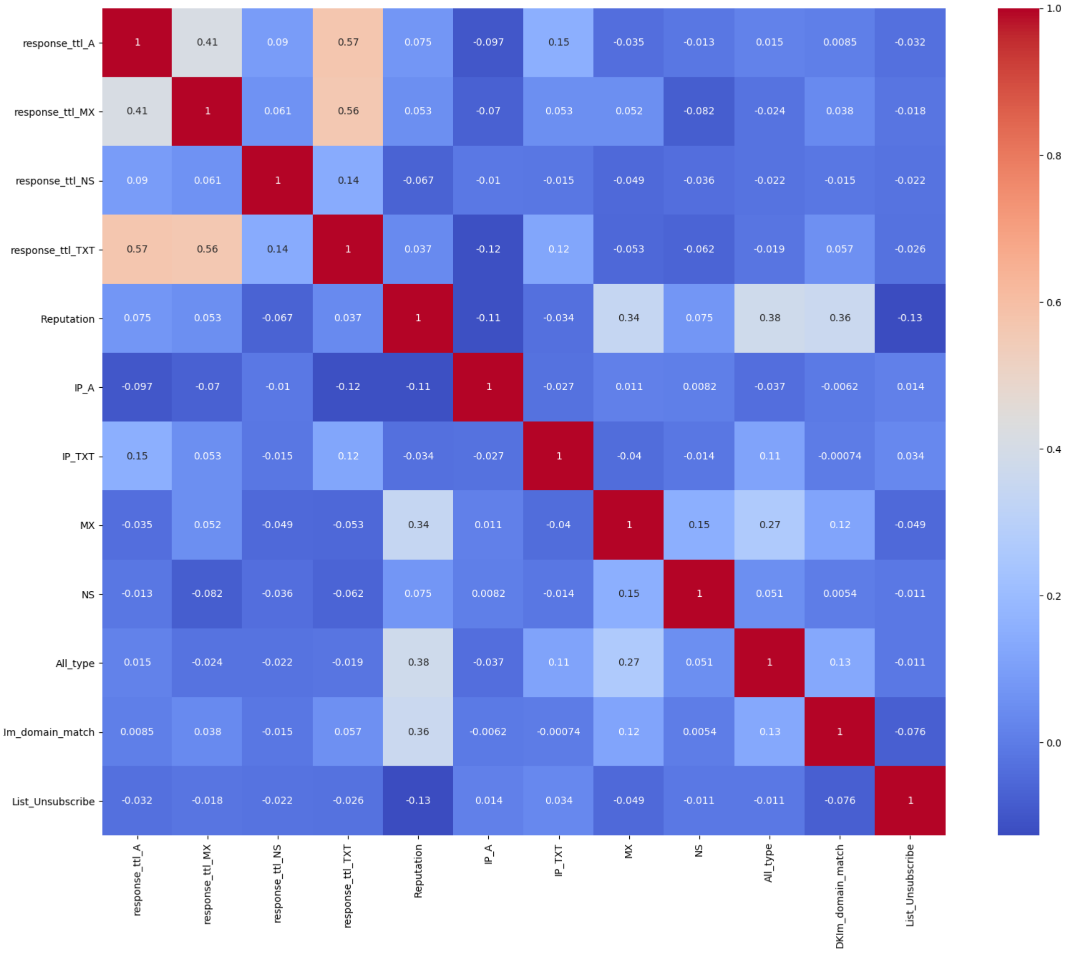
response_ttl_TXT	DNS Record	Numerical
Reputation	DNSBLs	Dichotomous
No. IP in A	DNS Record	Numerical
No. IP in TXT	DNS Record	Numerical
No. of MX Domains	DNS Record	Numerical
No. of nameservers	DNS Record	Numerical
All_type	DNS Record	Numerical
SPF_Include	DNS Record	Categorical
Dkim_domain_match	Email Header	Categorical
List_Unsubscribe_match	Email Header	Categorical

Table 1: Features Used

3.3 Data Preparation and Exploration

The quality of the output of the detection model depends upon how well training dataset is prepared. The active DNS records were in the form of AVRO format which is a JSON but is serialized and has a schema attached to it used for big data analytics **[17]**. Total dataset consisted of around 10 million records which could have been impossible to search for each domain in the email without using FASTAVRO **[18]** a python module for traversing avro files. Each of the avro record whether being ‘A’ or a ‘TXT’ record consisted of 114 key value pairs out of which only few were of our interest. Hence, we made use of Pandas library and pivoted the data according to our needs. Even after doing some bit of cleaning through automated process most of our records consisted string values which needed to be converted in to categorical or dichotomous variables, for that purpose we used some excel functions to further clean the data.

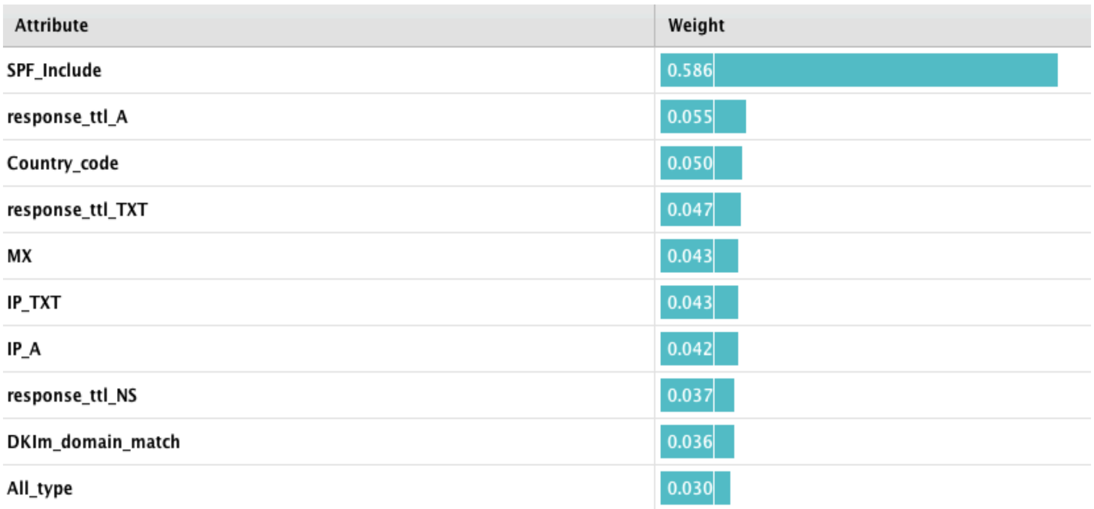
In data exploration, we analyzed the importance of each feature in our dataset. We drop ‘domain_name’ because it was a unique identifier. For missing or null values, we interpolated the data by filling them with mean of before and after of the missing values for that particular column. We chose interpolation because we know the pattern of the dataset therefore if we choose the mean of entire columns that could tamper with the standard deviation. We also converted our categorical variables into numbered columns so that they become discrete which is important at the time of model creation and then separated our predictor variable i.e. ‘Reputation’ on to other data frame. Correlation analysis was performed to find out whether the corresponding columns are highly correlated to each other or not and there was not much high correlation found in between the variables as seen from the figure below. Most of the boxes are highlighted in blue which shows very less collinearity between variables. As a result, there is no need to drop any of the features



	response_ttl_A	response_ttl_MX	response_ttl_NS	response_ttl_TXT	Reputation	IP_A	IP_TXT	MX	NS	All_type	DKIm_domain_match	List_Unsubscribe
response_ttl_A	1	0.41	0.09	0.57	0.075	-0.097	0.15	-0.035	-0.013	0.015	0.0085	-0.032
response_ttl_MX	0.41	1	0.061	0.56	0.053	-0.07	0.053	0.052	-0.082	-0.024	0.038	-0.018
response_ttl_NS	0.09	0.061	1	0.14	-0.067	-0.01	-0.015	-0.049	-0.036	-0.022	-0.015	-0.022
response_ttl_TXT	0.57	0.56	0.14	1	0.037	-0.12	0.12	-0.053	-0.062	-0.019	0.057	-0.026
Reputation	0.075	0.053	-0.067	0.037	1	-0.11	-0.034	0.34	0.075	0.38	0.36	-0.13
IP_A	-0.097	-0.07	-0.01	-0.12	-0.11	1	-0.027	0.011	0.0082	-0.037	-0.0062	0.014
IP_TXT	0.15	0.053	-0.015	0.12	-0.034	-0.027	1	-0.04	-0.014	0.11	-0.00074	0.034
MX	-0.035	0.052	-0.049	-0.053	0.34	0.011	-0.04	1	0.15	0.27	0.12	-0.049
NS	-0.013	-0.082	-0.036	-0.062	0.075	0.0082	-0.014	0.15	1	0.051	0.0054	-0.011
All_type	0.015	-0.024	-0.022	-0.019	0.38	-0.037	0.11	0.27	0.051	1	0.13	-0.011
Im_domain_match	0.0085	0.038	-0.015	0.057	0.36	-0.0062	-0.00074	0.12	0.0054	0.13	1	-0.076
List_Unsubscribe	-0.032	-0.018	-0.022	-0.026	-0.13	0.014	0.034	-0.049	-0.011	-0.011	-0.076	1

Figure 1: Correlation within Variables

Importance of each of the features was analyzed using random forest and Figure 3 shows the importance of features which will be best suited for our detection model. It was found that SPF_Include feature is most suitable because most the spammers do not add ‘include’ statement in their SPF record which helps them bypass the gateways having weak authentication.



Attribute	Weight
SPF_Include	0.586
response_ttl_A	0.055
Country_code	0.050
response_ttl_TXT	0.047
MX	0.043
IP_TXT	0.043
IP_A	0.042
response_ttl_NS	0.037
DKIm_domain_match	0.036
All_type	0.030

Figure 2: Feature Importance

However, other features did not show much of a spike in importance, so let’s examine the categorical ad numerical variables for reputation.

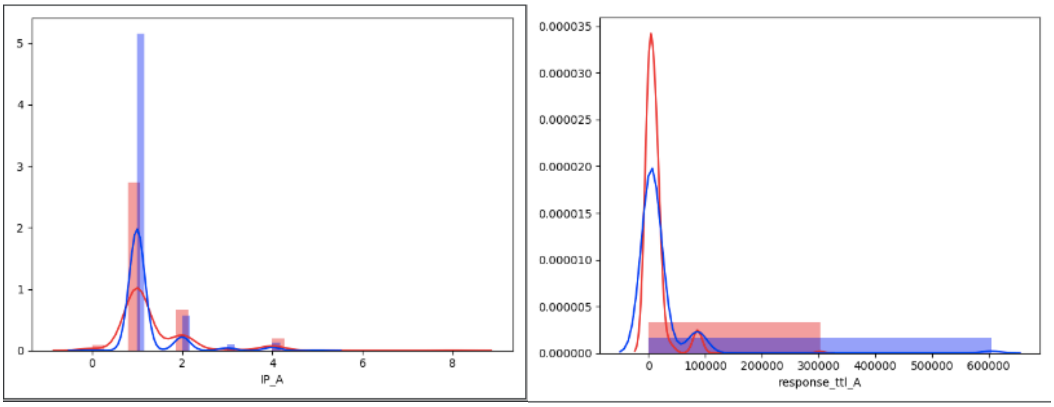


Figure 3: IP_A , TTLl_A Histogram

From the histogram on the left it can be seen that malicious domains(in red) have more group of IP addresses some malicious domains can have more than 2 IP addresses within same nameserver and on the right we have TTL of A record depicting comparatively short TTL values for SPAM domains and the graph is also positively skewed, some values could be outliers in this case.

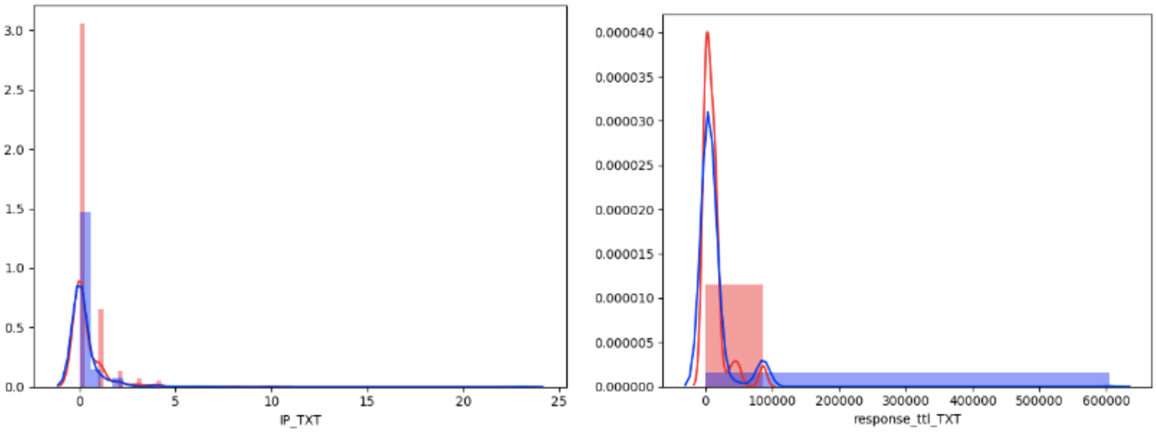


Figure 4: IP_TXT and TTL_TXT

Above graphs show that again there are number of IP addresses used by spammers for TXT record as well and there TTL values are also very low as compare to the benign domains, hence we should keep these features as well.

3.4 Classifier Selection

The models used till now for spam detection are old and mostly single layered. In order to select the best classifier for our problem we looked at and familiarized with Tree-type models Such as Decision trees, Random forest**[19]** along with Nearest neighbor**[20]** alternatives but we finally go with Multi level perceptron model used in **[5]** which is a deep learning algorithm. Although the other models are good for clustering but for classification or prediction, we require algorithms inspired by real world problems such as ANN (Artificial Neural Networks) containing multiple layers of networks of neurons **[15]**.

4. Design Specification

4.1 About the Classifier

This algorithm is very flexible, powerful and easily scalable and can manage big tasks of n number of dimensions with ease. These were earlier used in image recognition, drug-designs, language translations etc. They have not been used in SPAM detection earlier. Before neural networks, SVM (Support Vector Machines) were popular amongst predictive analysis which require high dimensionality, but success rate of SVM depends on highly feature rich dataset that to with the help of precise and correct data extraction.

The fundamentals of this algorithm are the neurons connected to each other through multiple layers and each layer can have large number of neurons to imitate the synapses just likex inside a brain. One of the benefits that make this algorithm more powerful is Backpropagation, in which the error of prediction is reused to change the weights of the neurons in previous layer and so on until it propagates back to input layer and thus improving accuracy of the entire network. One more benefit is that it allows stacking up of more layers to boost the potential of learning and increase the accuracy, this can possibly benefit because it assures model overfitting, adds up more time to train. Also, it has activation functions which determines when a neutron should be fire and also decides the size of output depending on previous layer input. They bring non-linearity and because of non-linearity neural networks can detect much complex patterns. This model is also well suited for our dataset which has very less rows as compared to large datasets can predict the result more accurately than any other model applied. Another great thing about this algorithm is that there are no assumptions to be followed and hence pattern of data is not important. Figure 5 shows the working of a deep neural network model where $\{i_1, i_2, i_3\}$ is the input layer which get values as $\{x_1, x_2, x_3\}$ and the hidden layer in the middle $\{h_1, h_2, h_3, h_4\}$ and finally an output layer $\{o_1, o_2, o_3\}$.

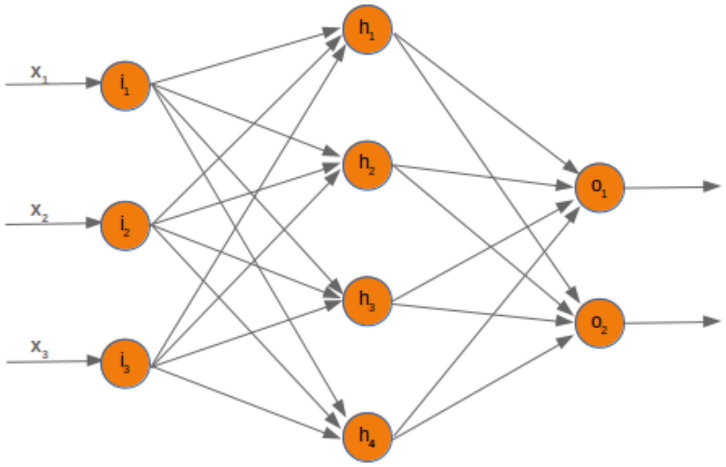


Figure 5: Neural Network Model

Each of the arrow shown above has a particular weight so with the input $\{x_1, x_2, x_3\}$ will be distributed from input layer to the hidden layer based on the weights assigned to each of the arrow and a weight matrix will be formed multiplied with each perceptron on hidden layer. The perceptron h_1 will compute $\O(W_{12},X_1 + W_{12},X_2 + W_{12},X_3)$ and so on and feed the calculation to the outer layer. Similarly, more layers can be added for in-depth calculations. In order to fit the data to the model before training we will scale the data, we split the data into training and evaluation set in a 60:30 split, drop our predictor variable and put it onto other axis, for evaluation of our results we will use F score, accuracy and recall and precision. From F score we will be able to get the degree of variation in between the training features whereas accuracy will help us know the percentage of difference between the predicted and actual value on the other hand recall will depict amount of actual positive samples determined correctly and precision will answer the percentage of positive case identified were really correct or not. We will compare our result of neural network model with a logistic regression model which will act as a baseline of accuracy for our model.

4.2 Model Overview

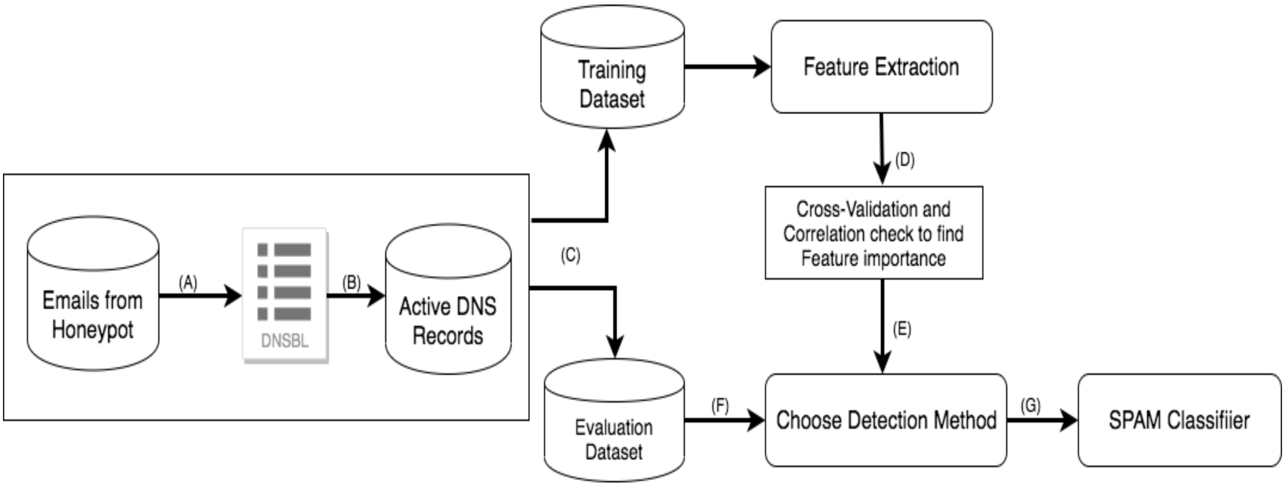


Figure 6: Proposed Methodology

To classify emails optimally within a business network Figure 5 shows the steps to create. From (A) to (D) it constitutes the detection process and (E)-(F) constitutes the classification process. In (A) the email data is hosted by **[3]** is in text format with all the email headers intact and is archived on monthly basis in a 7zip format. There is no lisence or ethics approval required to download the data. We only required the headers of the email to get such as DKIM signature domain and List-Unsubscribe from which we will derive DKIM domain match and List-unsubscribe domain match. Moving on there are two industry acclaimed and widely used DNSBLs such as SPAMHAUS and SURBL, in (B) we query the domain name to these blacklists to find its reputation whether it is a ham domain or a spam domain and based on that we find out the active DNS records from the OpenINTEL dataset which is also downloadable and does not require ethics approval for download. This data is in AVRO format from Apache foundation. For a corresponding domain name and we fetch the ‘A’, ‘NS’, ‘MX’ ‘TXT’ records. From those records further we fetch features listed in Table1. We build a master dataset with those features in (C) divide it into two sets for training and evaluation respectively. We analyze the features in (D) and their importance by various statistical research methods to find out the relevance of each feature. The detection model chosen in (E) was first baselined with a simple logistic regression model to achieve maximum accuracy above the base model. After baselining, MLP classifier was chosen and to achieve maximum results the classifier was executed in 3 ways :-

1. Without Backpropagation and Gradient Descent Solver – This process involved normal mapping and multiplication of weights with inputs. Backpropagation is the method from which we can minimize the loss function and move towards negative derivative which can result in better prediction and this can be done through gradient descent.
2. With best suited Gradient Descent and Solver – There are many sophisticated solvers available such as 'LBFGS', 'ADAM' and 'SGD' which are used for such optimizations. in results. We chose these three and compare the results which gives the best results. Along with the solver we used GD regularization coefficient to check whether the classifier is overfitting with our dataset or not.
3. With best suited Gradient descent and Solver along with addition of hidden layers – Before hidden layers we try to introduce batch sizes for better sampling and to remove any imbalance in data. Hidden layers are such hyper parameters which can change the topology or architecture of the neural network. Introduction of hidden layers increase the dimensional space for further classification of the result.

5. Implementation

The research was done on MAC OS X with Core i5 processor of clock speed 2.2 Ghz with 8 GB ram. Language used for programming is python and the IDE used was Pycharm. The reason for choosing Pycharm was that it has built-in support for data science such as Python dev console, Data view and Plots view for various statistical methods used.

5.1Data Collection and Cleaning

To extract email headers from the text files we used email parser library in python, which extracts all the key value pairs from headers, the number of keys present were not fixed as some of the emails came from a lot of relay servers so they consisted of lot of 'Received-From' headers which usually consisted of IP addresses of those servers. Emails from the month January to December of 2019 were fetched and parsed through email parser module and stored the respective headers onto a csv file. The received from field was not clean and consisted of username of the sender as well as some extra information. So, it was cleaned using regex pattern for domain name. Same was done to clean DKIM as well as List-Unsubscribe feature. For querying blacklists python has a library named 'spam-lists' **[21]** which can be easily installed through pip and contains classes for clients such as SURBL, SPAMHAUS, Google Safe Browsing. The lookup method recursive searches in SPAMHAUS and SURBL and returns a binary value i.e. for spam it returns TRUE and for ham it returns false. The csv file is updated with new column named 'Reputation' to store the result of blacklist query.

Then for active DNS records library named FASTAVRO **[18]** is used which provides fast fetching of results the using normal AVRO library because the normal library is written in pure python but FASTAVRO is written using some of the C extensions which makes iterating over the records much faster. The data dictionary of the records is very long and contains 114 key value pairs which means for each record be it 'A', 'TXT' or 'MX', 114 columns are returned. To overcome this problem, we used Pandas pivot and melt function to break down the structure and fetch only those columns which are of interest. The columns fetched are still not cleaned and ready for data exploration, hence some manual cleaning with the use of excel was done.

5.2 Classifier Implementation

Preprocessing and scaling of data were done before splitting the data to train and test.For changing categorical variable into corresponding hot values we use Pandas 'get_dummies' function. For checking feature importance, we used Random forest classifier and with the help of Matplotlib library we generated bar graph of the features in hierarchy of importance. Then firstly from Scikit Learn python library Logistic regression model was chosen which is our base classifier and trained the data and for calculation of scores we imported Scikit learn's metrics package which consists of accuracy_score, f1_score, classification_report and confusion_matrix. Matplotlib and Seaborn libraries were also used for generating correlation matrix of the features. For implementing our neural network classifier MLPClassifier was imported from Scikit learn's Neural Network package. At the first instance simply, the classifier was trained, and scores and the result were calculated. To further train the classifier we used gradient descent solvers as well introduce new layers as mentioned in Design Specification section to increase the accuracy of our results.

6. Evaluation and Discussion

We split the dataset into 60:30 ratio and start with applying the Linear regression model on to the features explored. For performance analysis of the results based on Accuracy, Precision, F1, Recall and also analyze the confusion matrix which will predict the actual and determined values. The Accuracy and precision can be calculated through the confusion matrix because it consists of all four values i.e. True Positives, False Positives, True Negative and False Negative. The formula for calculating Accuracy and Precision is as below.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

When there is a statistical analysis which involves classification in terms of binary values F1 score becomes really important when the class distribution is uneven which means large number of actual negatives. Formula for Recall and F1 are as below.

$$Recall = \frac{TP}{TP + FP}$$

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Classifier	Accuracy	Precision	Recall	F1
Logistic Regression	95.06%	93.02%	100.00%	96.39%

Table 2: Logistic Regression classifier results

Confusion Matrix

$$\begin{bmatrix} 71 & 12 \\ 0 & 160 \end{bmatrix}$$

The base model has quite decent accuracy, but the confusion matrix depicts that there are 12 false positives that were detected which is a large number for such small dataset. Also, precision is just 93.02 %. But since there are large number of actual negatives, we need to consider F1 score as well, which is also pretty decent.
Now moving on to first step of our MLP classifier we run the classifier with any GD solver.

Classifier	Accuracy	Precision	Recall	F1
Generic MLP	74.90%	79.2%	78.8%	78.89

Table 3: Generic MLP

Confusion Matrix

$$\begin{bmatrix} 68 & 29 \\ 32 & 114 \end{bmatrix}$$

The accuracy for the generic MLP is very less compared to our baseline model as well as the confusion matrix depicts around 29 false positive which is also a big number. So we moved further and try to introduce a GD solver which can

	Solver	Accuracy	F1
0	lbfgs	91.769547	93.75000
1	sgd	80.658436	85.714286
2	adam	71.604938	76.124567

Table 4: MLP with Solvers

It can be seen from the above table that 'lbfgs' solver gives out the maximum accuracy and F1 score. Hence, we moved in the right direction and try to add suitable activation function along with the lbfgs solver. We tried with 4 activation functions namely

	activation	Accuracy	F1	recall	precision
0	identity	87.242798	89.836066	86.708861	93.197279
1	logistic	93.004115	94.533762	93.037975	96.078431
2	tanh	87.242798	89.836066	86.708861	93.197279
3	relu	93.004115	94.670846	95.56962	93.78882

Table 5: MLP with Solver and Activation function

The classifier performs well with the use of logistic activation as seen from the above table and seems to improve the overall result. Furthermore, to improve the results from the classifier we can try to adjust the regularization coefficient to avoid overfitting and reduce the extra noise from our data.

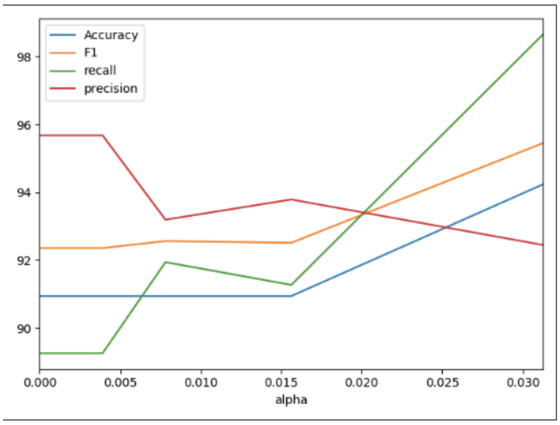


Figure 7:MLP with Regularization

Accuracy	F1	recall	precision	Alpha
92.18107	93.811075	91.139241	96.644295	0.03125

Table 6: Result from MLP with regularization

The results from adjusting the regularization coefficient can be seen in the above table and it can be suggested that performance has not been improved, in other words we can say that our data already has appropriate fitting, now we try to adjust the batch size along with the value of regularization coefficient. Batch size is again a hyper parameter which determines the number of samples to be used before updating the parameters of the internal model. Batch size can be very useful in our case because of our small dataset.

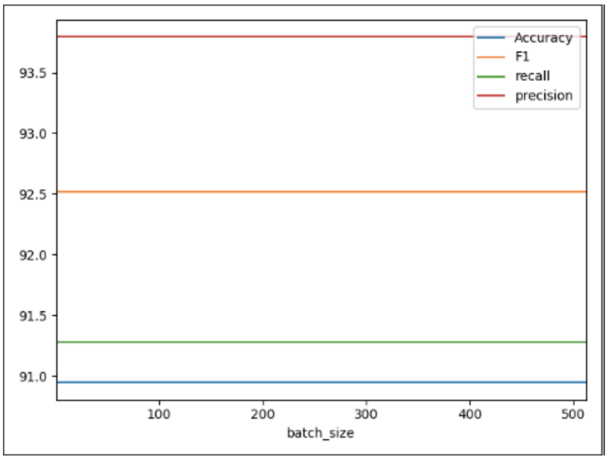


Figure 8: MLP with introduction of batch size

Our batch size seems to be constant throughout irrespective of changing the batch size and the table also tells that batch size does not affect the accuracy value. Hence. we will not include batch size and process with introducing hidden layers to see the improvement in the results. The main job of hidden layer here will be to transform intermediary inputs to something useful for the output layer hence advancing the simple Multi-Layer Perceptron to Feed Forward variant.

hidden_layer_sizes	Accuracy	F1	recall	precision
98 (8, 1)	97.119342	97.805643	98.675497	95.705521

Table 7: Feed Forward MLP with 2 hidden Layers

Looking at the result from above table by introduction of 2 hidden layers both having size of 98 represented as 8x1 matrix, we have reached the accuracy of 97% and with 95% precision which is a very good score. An F1 score of 97.8% depicts that our model has been learning well with a recall of 98.67%. Given a small amount of data this can be considered not best but a decent score which means that our feature selection i.e. inclusion of DKIM signature domain and List-Unsubscribe from the email has proven to be correct.

7. Discussion and Challenges

To detect spam and ham domains previous work make use of the Email reception log and Active DNS data provided through a private ISP, the amount of dataset was huge and contained more malicious domains and all of them were prelabelled. In our case we had to manually query the domains onto blacklists and then fetch their active DNS records. We have achieved almost the same accuracy as that of work **[13]** but our precision value is much better.

	Accuracy	Precison
[13]	97.11%	88.09%
Our Method	97.11%	95.70%

However, there can be various challenges that can be faced while implementation of this model in terms outcome because once a method predicts domain to be spam it does not tell which category the spam belongs to. The other challenge belongs to highly dynamic nature of spammers which constantly adapt and imitate a legitimate email behavior thus it becomes hard for capturing such spams. The spammers run a whole content delivery network of spams, but the difference is the bot masters have very loose control over the infected machines hence the uptime of the machines is very less as a result these spams become even more hard to catch hold of them and even they cant be taken down because of the large IP address pool of botnet.

8. Conclusion & Future Work

In this paper we discussed how with the use of both email characteristics as well as authoritative DNS records can help detecting the spam emails. We built a classification model predicting the domains with 97% accuracy, which can help the information security team to identify these emails in early stages and can save a lot of time and manual work of spam detection. This research can be implemented where the SMTP traffic is huge such as in Fortune 500 companies and just by merely applying regex patterns for already discovered spam pattern is not an efficient way considering such highly adaptive nature of spammers. Along with using the information from authoritative DNS servers can help in fighting against domain spoofing as the authoritative servers could reveal feature rich traces of the malicious domains which help track them and bring them down.

In future work we would like to work on a bigger and industry related email dataset from the likes of O365 exchange server with a longer log period in which we can also look upon social engineering characteristics of the mail.

9. References

[1]"RFC 6376 - DomainKeys Identified Mail (DKIM) Signatures", Tools.ietf.org, 2019. [Online]. Available: https://tools.ietf.org/html/rfc6376. [Accessed: 08- Jan- 2020]

[2]"RFC 7208 - Sender Policy Framework (SPF) for Authorizing Use of Domains in Email, Version 1", Tools.ietf.org, 2020. [Online]. Available: https://tools.ietf.org/html/rfc7208. [Accessed: 08- Jan- 2020]

[3]"SPAM Archive", Untroubled.org, 2019. [Online]. Available: http://untroubled.org/spam/. [Accessed: 08- Jan- 2020]

[4]"OpenINTEL: Active DNS Measurement Project", Openintel.nl, 2020. [Online]. Available: https://openintel.nl/. [Accessed: 08- Jan- 2020]

[5] O. van der Toorn, R. van Rijswijk-Deij, B. Geesink and A. Sperotto, "Melting the snow: Using active DNS measurements to detect snowshoe spam domains", NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium, 2018.

[6]L. Bilge, S. Sen, D. Balzarotti, E. Kirda and C. Kruegel, "Exposure", ACM Transactions on Information and System Security, vol. 16, no. 4, pp. 1-28, 2014.

[7]S. Hao, A. Kantchelian, B. Miller, V. Paxson and N. Feamster, "PREDATOR", Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security - CCS'16, 2016.

[8]S. Suwa, N. Yamai, K. Okayama and M. Nakamura, "DNS Resource Record Analysis of URLs in E-Mail Messages for Improving Spam Filtering", 2011 IEEE/IPSJ International Symposium on Applications and the Internet, 2011.

[9]"Deep Learning Based-Phishing Attack Detection", International Journal of Recent Technology and Engineering, vol. 8, no. 3, pp. 8428-8432, 2019.

[10]A. Qaroush, I. Khater and M. Washaha, "Identifying spam e-mail based-on statistical header features and sender behavior", Proceedings of the CUBE International Information Technology Conference on - CUBE '12, 2012.

[11]Y. Zhauniarovich, I. Khalil, T. Yu and M. Dacier, "A Survey on Malicious Domains Detection through DNS Data Analysis", ACM Computing Surveys, vol. 51, no. 4, pp. 1-36, 2018.

[12]P. Owezarski, "Unsupervised classification and characterization of honeypot attacks", 10th International Conference on Network and Service Management (CNSM) and Workshop, 2014.

[13]K. Dan, N. Kitagawa, S. Sakuraba and N. Yamai, "Spam Domain Detection Method Using Active DNS Data and E-Mail Reception Log", 2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC), 2019.

[14]"The Scuffle Between Two Algorithms -Neural Network vs. Support Vector Machine", Medium, 2019. [Online]. Available: https://medium.com/analytics-vidhya/the-scuffle-

between-two-algorithms-neural-network-vs-support-vector-machine-16abe0eb4181. [Accessed: 08- Jan- 2020]

[15] A. Zeichick, A. Zeichick, Akrepburcu, Kelson, G. Galloway, LindaB, M. C, A. Ludgate, L. W. McDonald, Reg, Deborah, Loretta, Jim, E. J, K. Arbuckle, Nigel, Mark, M. Stockley, K. Phillips, P. Ducklin, BobR, Michael, Scott, Enzo, Anon, N. Gray, C. Simpson, Paulette, S. R, J. R, J. Norrell, E. Lock, P. Ducklin, L. Vaas, L. Vaas, M. Stockley, and Jk, "5 things you should know about email unsubscribe links before you click," Naked Security, 04-Sep-2014. [Online]. Available: https://nakedsecurity.sophos.com/2014/09/04/5-things-you-should-know-about-email-unsubscribe-links-before-clicking/. [Accessed: 08-Jan-2020].

[16] "What is Round-Robin DNS and how to set it up?," ClouDNS. [Online]. Available: https://www.cloudns.net/wiki/article/182/. [Accessed: 08-Jan-2020].

[17] "Welcome to Apache Avro!," Welcome to Apache Avro! [Online]. Available: https://avro.apache.org/. [Accessed: 08-Jan-2020].

[18] Fastavro, "fastavro/fastavro," GitHub, 20-Dec-2019. [Online]. Available: https://github.com/fastavro/fastavro. [Accessed: 08-Jan-2020].

[19] Antonakakis, M., Perdisci, R., Dagon, D., Lee, W., & Feamster, N. (2010). "Building a Dynamic Reputation System for DNS. USENIX Security Symposium".

[20] Feamster, Nick & Gray, Alexander & Krasser, Sven & Syed, Nadeem. (2008). "SNARE: Spatio-temporal Network-level Automatic Reputation Engine". Georgia Institute of technology
[21] DBL - The Spamhaus Project. [Online]. Available: https://www.spamhaus.org/dbl/. [Accessed: 08-Jan-2020].