International Journal of Advanced Science and Technology
Vol. 29, No. 5, (2020), pp. 5165 - 5176

The Exposing Deleterious Websites on Domain Name System

P. Varaprasada Rao[1], P. Chandra Sekhar Reddy[2], S. Govinda Rao[3], Y.Manoj Kumar[4], G.Anil Kumar[5]
[1,2,3] *Professor,*[4,5] *Assistant Professor Computer Science and Engineering, GRIET, Hyderabad, India.*

Abstract

With the developing utilization of the web over the world, the dangers presented by it are various. The data you get and share over the web is open, can be followed and altered. Pernicious sites assume a crucial job in affecting your framework. These sites arrive at clients through messages, instant messages or wicked notices. Consequence of these destinations would as often as possible be a downloaded malware, spyware, ransomware and traded off records. Malevolent site updates activity on clients' location in any case, on account of drive-by just downloads, site will endeavor to introduce programming on PC not taking consents from customers approval first. Set forward a model to figure a URL is malignant or benevolent, in light of the application layer and system attributes. AI calculations for arrangement are utilized to build up a classifier utilizing the focused-on dataset. The concentrated-on dataset is isolated into getting ready and endorsement sets. These sets are used to plan and support the classifier model. The hyperparameters are tuned to refine the model and make better results.

Keywords:*Websites,Applicationlayer, Network characteristics, Machine learning, Classification, Classifier, Hyperparameters*. **.**

I. INTRODUCTION

The development of new correspondence advances has hugy affected the development of organizations over numerous applications including web-based banking, online business and long-range interpersonal communication. In reality, in this day and age, an individual should be online for having a prosperous vocation. The noteworthiness of the web is developing ceaselessly. Nonetheless, the progression in advances has accompanied dangers (cybercrime) that put its clients into a terrible circumstance. Such perils consolidate dissident security programming, adware, spyware, phishing, DOS and DDOS. These risks reliably lead to abuse of customers' information and wealth. A structure that isn't guaranteed using reasonable security controls can be sullied with threatening programming anyone can burst into your system and access your data. Noxious URLs and tainted sites are the main driver of such dangers. A vindictive URL is an undermined URL which frequently prompts downloaded malware, spyware, ransomware and traded off records.

The internet is built on is a global computer network providing various facilities such as communication, entertainment, business, so on. It is also known as an interconnected network. The Web grant the users to access data through URLs. These URLs are verified using DNS. It helps the URL to be converted into IP address.
Domain Name System (DNS) is a progressive and decentralized naming framework which permits us to associate with the web. Each device connected to the internet has a unique IP address which enables other machines to find the device. This makes users access URLs without remembering the IP address (IPv4 or IPv6). URL is a one of a kind identifier used to find an asset over the web as is characterized as the worldwide location of reports and different assets on the internet. A URL commonly contains two sections convention name and an area name or IP address. The name of the convention is expected to get to an asset and asset name. It chooses the convention to be utilized while getting to assets. Space and subdomain are utilized to find the resource. Malicious URLs represent 33% of complete URLs that exist. Innocent clients of the web are the essential focuses of such locales. Most continuous assaults will in general be driven by download, phishing and social designing and spam. A drive-by download is a hurtful programming code that is introduced on a client's framework without the client's consent. Phishing and social
building assaults use bargained sites to look for individual data by going about as a reliable association. Spam can be a hazardous wellspring of the danger and is a piece of the phishing trick.

ISSN: 2005-4238 IJAST

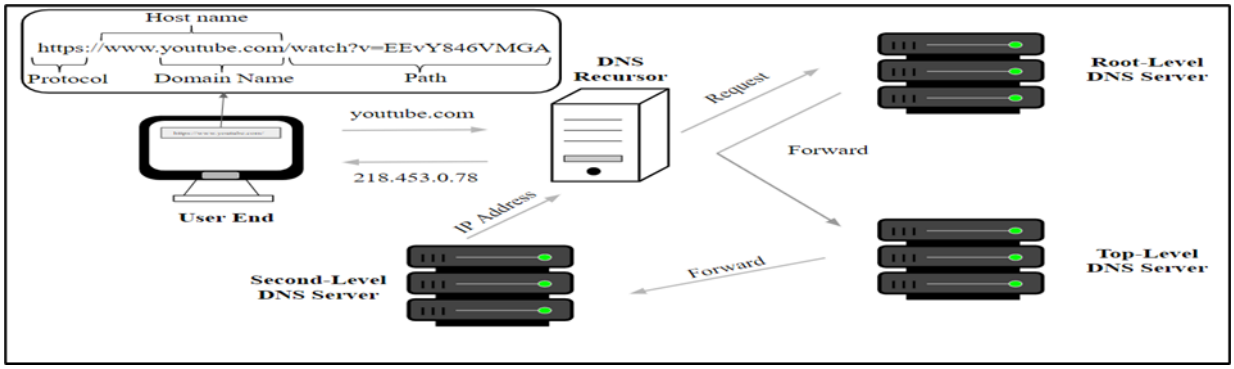

Figure 1. The DNS Architecture

The most widely recognized technique utilized by antivirus bunches is boycotting strategy. It utilizes a database where every malevolent Url accessible and distinguished are put away. This database is refreshed every now and again when another vindictive URL is found. Aggressors are inventive enough to conquer the boycotted URLs and produce URLs which have all the earmarks of being authentic. Probably the most well-known approaches to distinguish a URL is undermined or not is utilizing an IP address, long or short URLs, sub and multi-areas, space enrolment, nonstandard ports, spring up windows, age of the area. These aggressors utilize an algorithmic way to deal with create URLs which make them look genuine and are not boycotted out. Subsequently, boycotting techniques have intense restrictions, and it is immaterial to conquer them, as boycotts are futile at whatever point another URL is produced. Attempted to advance different plans that are viable answers for distinguishing such URLs.

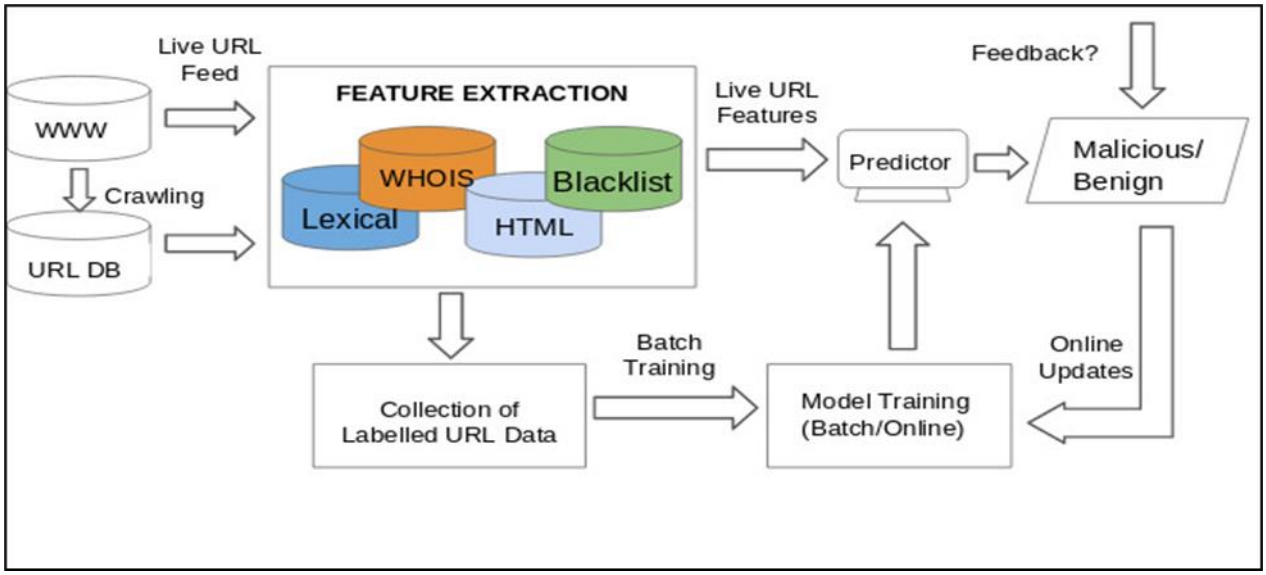

Figure 2. Architecture of Malicious URL detection

To beat these downsides, analysts have considered AI as a superior methodology. AI is utilized to foresee when the examples are covered up inside the dataset to make important expectations. The most widely recognized strategy for design revelation is design characterization.AI is characterized into administered, solo, semi-managed and strengthened learning dependent on the accessibility of preparing information.In supervised learning involves the learning in which we train machine using a labelled dataset and test with data that fall into these groups. For unsupervised learning considers data which is neither classified nor labelled.Reinforcement learning tries to maximization of the reward function. Here the agent decides what actions are needed to achieve a task.

Malicious URL detection process includes:

1.Feature Representation: Extracting the appropriate element portrayal: ut → xt where xt ∈ Rd (d-dimensional component vector speaking to URL).

2.Machine Learning: Learning a forecast work f: Rd → R which predicts the class task for any URL case x utilizing suitable component portrayals.

The dataset is gathered from Kaggle archive of malevolent and amiable URLs. This dataset is a regulated one marked as noxious and favourable.It is handled to evacuate NULL qualities and interject the information. The handling is done dependent on the properties gathered. In a dataset, all the properties are don't assume an unequivocal job in structuring the model. Subsequent to recognizing those properties and pre-preparing of information is done a productive AI calculation is considered for building up the model. Here our essential spotlight is on parallel grouping. In arrangement, we bunch information into a class and afterward anticipate the yield. Paired arrangement is the most essential method for order where we bunch things as a positive and a negative grouping. For distinguishing proof of pernicious URL, we describe URLs into dangerous or kind URLs. Thusly it would predict whether a URL is poisonous or not.Examination is planned for building up an AI model which groups the URLs. A dataset from Kaggle archive is utilized to fabricate the information outline for the model. This dataset is pre-handled and arranged for building the model. We actualize this model utilizing Extreme Gradient Boosting (XGBoost) classifier which is a troupe student. Ascertain the commitments made by the qualities and the precision picked up by every single one of them. Additionally, see the future upgrades as made for better outcomes. We centre around the modules that are to be actualized and the calculation used to assemble the model. We additionally centre around the highlights that help the model to be increasingly precise.

II. STATEMENT OF THE PROBLEM

The malevolent URL identification is a twofold grouping for forecast of noxious and kind.Dataset T URLs{(u_1, y_1), ..., (u_t, y_t)}, where u_t for t = 1, ..., T speaks to a URL from the preparation dataset, and $y_t \in \{0,1\}$ is the the comparing name which speaks to noxious or benevolent URLs.
The guideline focus of AI for harmful URL acknowledgment is to help farsighted exactness. Both the segment depiction and AI are basic to achieve this objective. While the underlying section of feature depiction relies upon space data and application layer, the later spotlights on setting up the request model by methods for a data driven improvement approach.

Figure 2 illustrates the architecture of solving the problem of detection of Malicious URL using a machine learning algorithm.
Acquiring the element vector x for the preparation information, to get familiar with the expectation work f: $R^d \rightarrow R$, it is frequently defined advancement issue to an extent that the

precision is amplified. The function used in the model is an objective function. It includes both training loss and regularization. The loss of training data decides how accurate the model

prediction will be and regularization decides the complexity of the model built. The objective function is defined as:

$$Obj(\Theta) = L(\Theta) + \Omega(\Theta)$$

Where we have training loss and regularization functions to define the objective.

Features of url:

The URL contains the domain name, protocol needed to access a resource, including its name. This is linked to information regarding the domain linked to the URL. This domain information is stored and accessed using DNS. It includes all the information such as registration date, port, its length, etc. We use only the necessary features for building the model. These features are selected using feature selection. Feature assurance and its generator are done by honeypot and PyShark. Each URL was executed on honeypot for 4 seconds with its default course of action and in equivalent, to the technique, a substance gets the web traffic through PyShark. Another mechanical assembly was made in Python for planning HTTP substance and WHOIS properties.

Accompanying attributes were created utilizing honeypot and PyShark:

ISSN: 2005-4238 IJAST

- URL: The Uniform Resource Locator of the website being analyzed for classification.
- URL_LENGTH: The length of the specified URL.
- NUMBER_SPECIAL_CHARACTERS: The number of characters in the URL which belong to the Special characters set {“/”, “%”, “#”, “&”, “. “, “=”}.
- CHARSET:Value denoting the character encoding standard.
- SERVER: An absolute worth meaning the working arrangement of the server.
- CONTENT_LENGTH: The length of the HTTP header's content.
- WHOIS_COUNTRY: A categorical value denoting the country obtained from WHOIS API.
- WHOIS_STATEPRO: A categorical value denoting the states obtained from WHOIS API.
- WHOIS_REGDATE: Denotes the server registration date and time format.
- WHOIS_UPDATED_DATE: The most recent update date of the server examined.
- TCP_CONVERSATION_EXCHANGE: The quantity of TCP parcels traded between the server and the honeypot customer.
- DIST_REMOTE_TCP_PORT:
  The quantity of unmistakable TCP ports identified.
- REMOTE_IPS: The all out number of IPs associated with the honeypot..
- APP_BYTES: The quantity of bytes moved over the system.
- SOURCE_APP_PACKETS: The packets sent from the honeypot to the server using the network.
- REMOTE_APP_PACKETS: The packets received from the server to the honeypot.
- APP_PACKETS: Number of IP packets generated during the honeypot and the server communication.
- DNS_QUERY_TIMES: Number of DNS packets generated during the honeypot and the server communication.
- TYPE: A categorical variable to specify the type of the website analyzed: 0 for benign, 1 for malicious.

DOMAIN_NAME	WHOIS_COUNTRY--IN	WHOIS_COUNTRY--IT	WHOIS_COUNTRY--JP	WHOIS_COUNTRY--KR
wikia.com	0	0	0	0
healthgrades.com	0	0	0	0
kfdm.com	0	0	0	0
labradorwest.com	0	0	0	0
kusports.com	1	0	0	0
amazon.com	0	0	0	0
usedregina.com	0	0	0	1
nj.com	0	0	0	0
waatp.com	0	0	0	1
healthgrades.com	0	0	0	0
greenwood-centre-hudson.org	0	0	0	0

Figure 3. One Hot Encoding on WHOIS_COUNTRY attribute

These features are recorded and uploaded to a dataset. A honeypot is a computer that acts as a likely target of cyber acts. Multiple honeypots can be created on a single network. PyShark is a wrapper for tshark. It allows packet parsing with the help of Wireshark dissectors.It is opened on a honeypot and its features are extracted using PyShark.

III. MODULES

Strategy:

Idea of the problem is formed the targeted dataset is constructed. Once the data collected it is then pre-processed. Then we use the dataset is split into test and train sets. Once we have the training set the model is built on it and used to test. This determines the accuracy of the model. The accuracy tells how well the model can predict the data.To impellent

this we split the whole process into sub-tasks. Each task is associated to a module that were implemented. It involves three phases namely data pre-processing, data splitting and modelling.

Dataset pre-processing:

Information pre-preparing is the significant advance in building up a model. It shapes the information for proficient expectations. This procedure incorporates information designing, cleaning and inspecting of information.
This process involves removal of NULL values, categorical values and unnecessary data or attributes. We eliminate NULL values using the concept of interpolation of the data. This concept involves the process of finding the mean of data and replacing the NULL value with the nearest value or mean based on the functions used.
To remove categorical values one hot encoding is utilized. It changes over clear cut qualities into a structure that a machine gets it. Because of this procedure, the machine sees effectively and builds up a model that gives progressively precise outcomes. It by and large proselytes the information into zero's and one's.
One hot encoding is the best way to remove such categorical values. In this process, we create dummy features representing these values. Once we generate dummies, we fill the dataset with binary values (machine-understandable language).

Dataset splitting:

The all the more preparing information is utilized, the better the potential model performs. Subsequently, more outcomes from model testing information lead to all the more likely model presentation and speculation ability.In some cases, we observe that the training set is biased. This makes the model prediction also influenced by the majority class. So, remove this problem we use the concept of oversampling and under sampling. We generate the data synthetically so that we have an equal amount of data for each class.Note that we do this on training data but not on the test data. It important to build an unbiased model but it's not necessary to test using the same kind of dataset.

Tree Building Algorithm

1.Iterate for n steps times

1.1 Grow the tree to the greatest profundity.

1.1.1 find the best parting point.

1.1.2 Assign a load to the leaf hubs.

2.Prune the tree to erase hubs with negative addition.

Modelling:

As the data considered is labelled, we use Supervised learning for recognizing a URL is noxious or not. Regulated techniques endeavour to find the relationship or examples between input qualities and a focused-on characteristic. The relationship or example found is spoken to in a structure alluded to as a model (tree, bunches, and so on). It tends to be finished utilizing characterization and relapse.
Once we decide the algorithm, we build the model on the generated training dataset. Boosting methods are used to improve predictions. XGBoost is one such boosting method.This model is tested using a test set which gives the accuracy of the model. Once we have the model, we find the importance of the feature to tune the parameters. This enables to increase the accuracy of the model. Also, the model can be trained on the input is given when used in applications.

IV.ALGORITHM

XGBoost is a decision-tree-based ensemble Machine Learning algorithm which uses a gradient boosting technique. When medium structured or tabular data is used, decision tree-based algorithms are considered to be the best in class right now. XGBoost algorithm is an optimized gradient boosting algorithm with parallel processing, tree pruning, handling missing values of data and regularization for avoiding overfitting. The algorithm is used in a wide range of applications and is also portable. We can integrate cloud with it (supports AWS, Azure, works with Spark, Flink ecosystems).

It is an implementation of gradient boosting. Some of the key features of this algorithm are:

- **Regularization**: Helps to prevent overfitting.
- **Handling Spare Data**: It uses the sparsity aware split finding algorithm to handle sparse data.
- **Weighted Quantile Sketch**: It has an algorithm to handle weighted data.

- **Parallelized Tree Building**: It can use multiple cores of the system as it has a square structure in its plan. It utilizes equal strings for checking and arranging circles.
- **Cache Awareness**: It allots inner cradles to each string, where the angle insights can be put away.
- **Out of Core computation**: To optimize the available disk space by handling the big data frames that don't fit in the memory.

To understand this boosting algorithm, Suppose we have **K** trees, then model generated is

$$\sum_{k=1}^{K} f_k$$

Where each f_k is the prediction from each decision tree. The model is built on a collection of decision trees. After generating decision trees, we make a prediction.

$$\widehat{y_i} = \sum_{k=1}^{K} f_k(x_i)$$

Where x_i is the feature vector of the i^{th} data point. Similarly, the prediction at the t^{th} step can be defined as

$$\hat{y}_i^{(t)} = \sum_{i=1}^{t} f_k(x_i)$$

To prepare the model the misfortune capacity ought to be streamlined. We use Log Loss for paired arrangement .

$$L = -\frac{1}{N}\sum_{i=1}^{N}(y_i \log(p_i) + (1 - y_i)\log(1 - p_i))$$

Regularization is a vital piece of a model. A decent regularization controls the multifaceted nature and overfitting of a model.

$$\Omega = \Upsilon^T + \frac{1}{2}\lambda\sum_{j=1}^{T} w_j^2$$

Where T is the number of leaves, and w_j^2 is the score on the j^{th} leaf.
When loss function and regularization function are put together, we get the objective of the model

$$Obj = L + \Omega$$

Misfortune work controls the prescient influence, and regularization controls the effortlessness of the model.

Goal Obj (y, ŷ) to streamline, XGBoost is an iterative system which computes

$$\partial_{\hat{y}}\, Obj(y, \hat{y})$$

At every cycle, we improve ŷ along the heading of the slope to limit the goal. We as certain second-request Taylor estimation for it and we get

$$Obj^{(t)} = \sum_{i=1}^{n}\left[g_i f_t(x_i) + \frac{1}{2}h_i {f_i}^2(x_i)\right] + \Omega(f_t)$$

This is the objective of the i^{th} step. Our goal is to find an f_t to optimize it.
The tree-building algorithm is built on two core concepts:

- **Internal Nodes**: Splits the flow of data points using a feature.
- **Leaves**: Each leaf has a weight. These weights represent the prediction.

It has the best prediction performance and processing time compared to other algorithms.
Implementing XGBoost (gradient boosting algorithm) does not ensure the best results. It is also important to choose the required configuration of the algorithm according to the dataset by tuning the hyperparameters.

V. RESULTS

To understand a dataset, it important to visualize its data points. This can be done using various methods such as box plotting, histograms, bar graphs, line charts and so on. These visualizations are provided by Matplotlib, Plotly, Seaborn, ggplot, Altair. Here Matplotlib, Seaborn is used for generating these visualizations.
When the dataset was put into a bar graph using Matplotlib we found that the dataset was biased. It had more data points which were under malicious class.

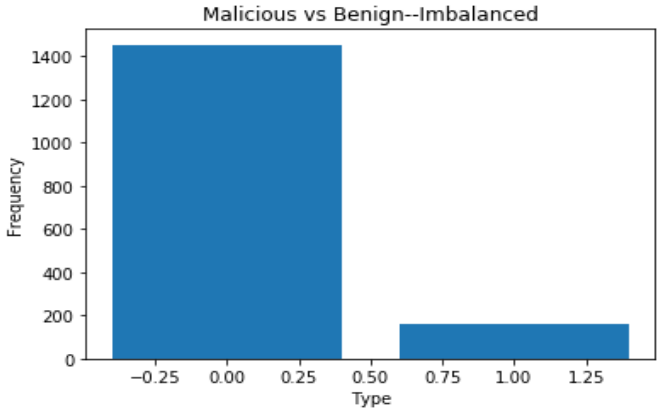

Figure 4. Malicious Vs Benign-Imbalanced

To overcome the problem of biasing we could perform oversampling and under sampling. This process is generally done on a train set. We do this only on train set because it improves the accuracy of prediction (and not on the test set because it would manipulate the original data and produce inappropriate results). In oversampling we add data points to the class with a lesser amount of data. This could be done by using approximate values which act as original values. These are generally known as dummy values.

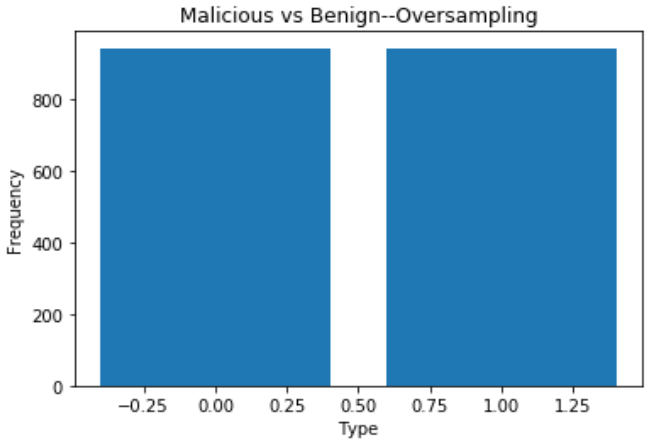

Figure 4. Malicious Vs Benign-Oversampling

A disarray framework is a table that is utilized to depict the presentation of an order model on the test set for which genuine qualities is called or characterized.

Comprises a accompanying arrangement of qualities:

- **True Positive (TP)**: Observed worth is certain and is anticipated to be sure.
- **False Negative (FP)**: Observed worth is certain, however is anticipated to be negative.
- **True Negative (TN)**: Observed worth is negative and is anticipated to be negative.
- **False Positive (FP)**: Observed worth is negative, however is anticipated to be certain.

Table 1. PredictedTable

	Predicted (No)	Predicted (Yes)
Actual (No)	489	23

Actual (Yes)	15	37

Characterization Rate or Accuracy is given by the connection::

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

Accuracy = 0.9326
The recall is defined as:

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

Recall = 0.711
High Recall shows the class is effectively perceived.Precision is defined as:

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

Precision = 0.616
F-measure uses Harmonic Mean in place of Arithmetic Mean as it punishes the extreme values more.

$$F - measure = \frac{2 * Recall * Percision}{Recall + Percision}$$

F-measure = 0.66

Table 2. Class Table to calculate Average

Class	Precision	Recall	f1-score	Support
1	0.97	0.96	0.96	512
0	0.62	0.71	0.66	52
Avg/Total	0.94	0.93	0.93	564

We can also analyse each feature of a URL with the output feature (Malicious or Benign). To do this we use the concept of the boxplot. In boxplot, we represent the data using minimum, quartile (Q1), median, third quartile (Q4) and maximum. Here we group data into quartiles and constructs box around them. In our dataset, we considered URL_LENGTH attribute with TIPO attribute to detect the outliers and the range of data points using the boxplot concept. The outlier data points are also called as abnormal points. These occur only for a few data points in the Dataset.

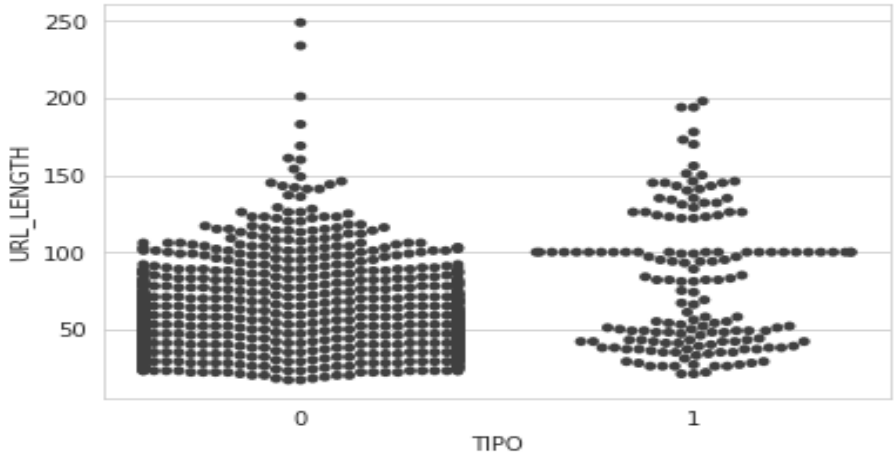

Figure 5. Each dot in the above graph represents the data points of the dataset.

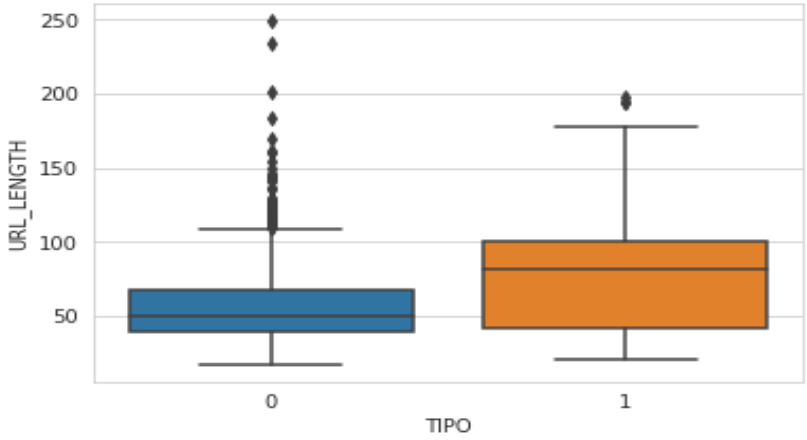

Figure 6. Url Length Vs TIPO

The diamonds above the boxplots are the outliers of the dataset. This clearly shows that these data points are not helpful to generate accurate results. But there will always be scenarios where we should consider these values too.
To understand the relationship of each attribute with the other attributes, we use the concept of correlation. In this concept, we generate the percentage of relationship with each attribute ranging from 0 to 1. To depict this relationship, we use a heat map.

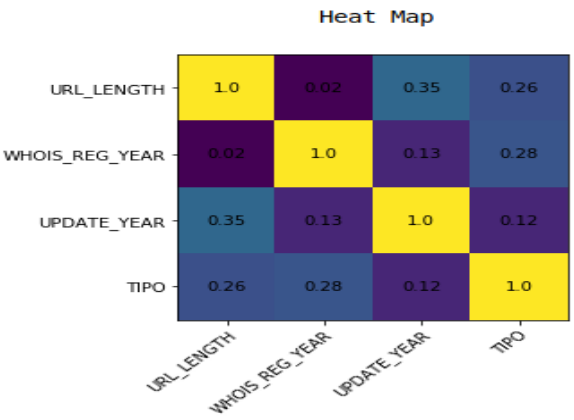

Figure 7. Heat Map

In this map, we see that each attribute related to itself is filled with 1 (as the attribute variates exactly same for itself).AUC-ROC bend is utilized to quantify the presentation of order at various limit esteems. ROC bend is a likelihood bend and AUC bend speaks to degree or proportion of distinguishableness. Higher the AUC, better the model is at recognizing pernicious and benevolent URLs.

The chart underneath speaks to the AUC-ROC bend for the classifier generated for classifying malicious and benign using XGBoost algorithm.

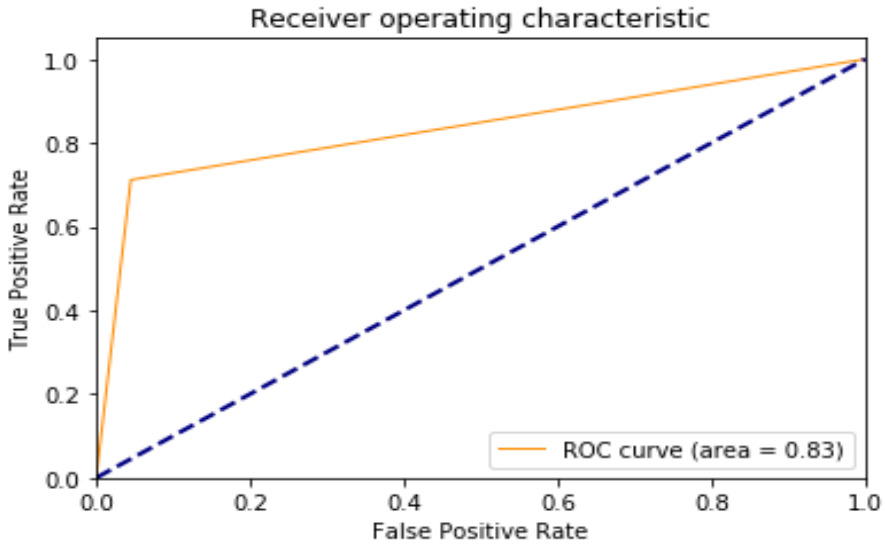

Figure 8. The region under the curve is AOC and the curve represents ROC.

Feature importance helps to identify useful features which contribute more weight while generating a model. It can be used for removing attributes which do not contribute to the development of the model. This also helps in understanding the model structure.

In feature importance, higher scores are having more hugeness or congruity of feature towards your yield variable. The significance of highlights for the dataset are:

Table 3. Attributes and ScoresTable

Attributes	Scores
NUMBER_SPECIAL_CHARACTERS	0.15879828
URL_LENGTH	0.13304721
REMOTE_APP_PACKETS	0.09656652
SERVER	0.08798283
DIST_REMOTE_TCP_PORT	0.072961375
CHARSET	0.06866953
WHOIS_COUNTRY	0.057939917
WHOIS_STATEPRO	0.05364807
DNS_QUERY_TIMES	0.051502146
CONTENT_LENGTH	0.047210302
SOURCE_APP_BYTES	0.03862661
REMOTE_IPS	0.03218884
APP_BYTES	0.025751073
WHOIS_REGDATE	0.025751073
REMOTE_APP_BYTES	ES 0.023605151

After deleting the attributes and generating the model the attributes are:

Table 4. Attributes and Scores Table

Attributes	Scores
DOMAIN_NAME	0.34332427
WHOIS_COUNTRY	0.31062675

ISSN: 2005-4238 IJAST

WHOIS_STATE_CITY	0.15803815
WHOIS_REG_YEAR	0.108991824
URL_LENGTH	0.065395094
UPDATE_YEAR	0.013623978

The feature importance graph is shown below:

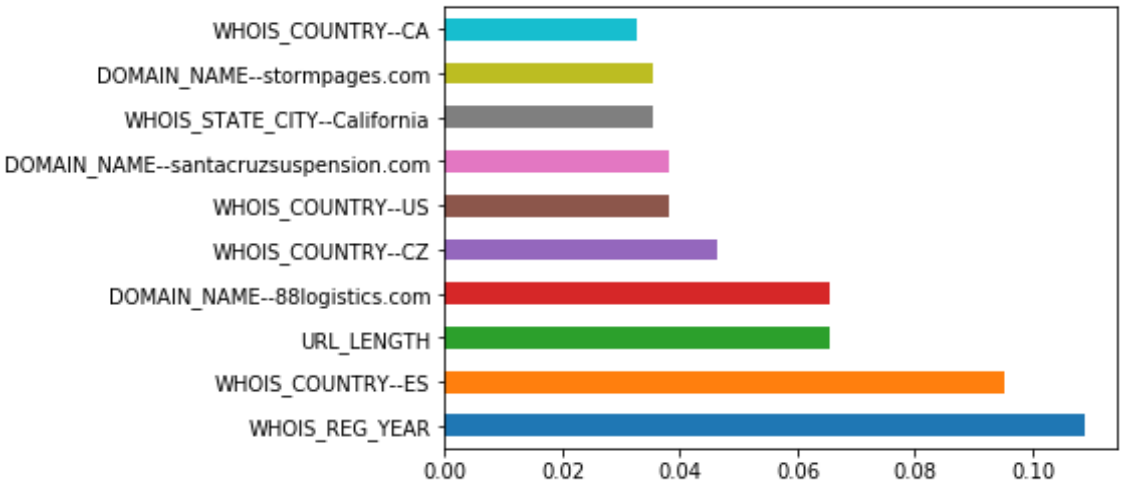

Figure 9.Graph represents the feature importance of dataset after one-hot encoding.

VI.CONCLUSION

The Dangers of metonym diseases via vindictive URLs are expanding nowadays, so activities are expected to forestall them. As a countermeasure against malignant URLs, boycotting URLs or areas are one among them. Yet, methods don't anticipate if a URL isn't in the database and likewise requires a tremendous measure of database for putting away completely boycotted URLs. Thus, to conquer this AI is a productive method to anticipate a URL as noxious or kind. Most ideal approach to utilizing arrangement calculations. Order URLs as malevolent or considerate, sort of grouping is called paired arrangement. Actualized utilizing choice trees, arbitrary woods calculation, Bayes hypothesis and some other characterization calculation.

The most ideal approach to actualize is utilizing XGBoost calculation is a group of students. This urges us to diminish the disaster and regularize the decision tree created at every movement of the count. One key segment of this endeavor is that we use one-hot encoding to change obvious characteristics to machine-sensible characteristics. Redesigns the speed of building the model and foreseeing it. In spite of the fact that the forecast is finished boycotting is as yet utilized as it is quicker when contrasted with any AI calculation and is increasingly exact a specific URL in dataset.

Simulated intelligence techniques are extending and their impact on different fields of building is in like manner more. One way to deal with improve the pace of desire and precision is to have more features and consider features with the most raised need and building the classifier again. This should be conceivable using feature noteworthiness.

Visual parts request by building a colossal dataset of threatening locales to cross-check against, future acknowledgment and portrayal can be improved by separating the visual segments which are shared among such destinations. This could be loosened up to DOM examination.

VII.REFERENCES

[1] P. Prakash, M. Kumar, R. R. Kompella, M. Gupta, "Phishnet: predictive blacklisting to detect phishing attacks", *INFOCOM 2010 Proceedings IEEE.*, pp. 1-5, 2010.

[2] P. de las Cuevas, Z. Chelly, A. Mora, J. Merelo, A. Esparcia-Alcazar, "An improved decision system forURL accesses based on a rough feature selection technique" in Recent Advances in Computational Intelligence in Defense and Security, Springer, pp. 139-167, 2016.

[3] A. Mora, P. De las Cuevas, J. Merelo, "Going a step beyond the black and white lists for URL accesses in the enterprise by means of categorical classifiers", *ECTA*, pp. 125-134, 2014.
[4] M.-Y. Kan, H. O. N. Thi, "Fast webpage classification using url features", *Proceedings of the 14th ACM International Conference on Information and knowledge management*, pp. 325-326, 2005.
[5] E. Baykan, M. Henzinger, L. Marian, I. Weber, "Purely URL-based topic classification", *Proceedings of the 18th International Conference on World wide web.*, pp. 1109-1110, 2009. [6] J. Ma, L. K. Saul, S. Savage, G. M. Voelker, "Beyond blacklists: learning to detect malicious web sites from suspicious urls", *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge discovery and data mining.*, pp. 1245-1254, 2009.
[7] "Learning to detect malicious URLs", *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, pp. 30, 2011.
[8] P. Zhao, S. C. Hoi, "Cost-sensitive online active learning with application to malicious URL detection", *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge discovery and data mining.*, pp. 919-927, 2013.
[9] J. Ma, L. K. Saul, S. Savage, G. M. Voelker, "Identifying Suspicious URLs: An Application of Large -scale Online Learning", *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 681-688, 2009.
[10] R. O. Duda, P. E. Hart, D. G. Stork, Pattern classification, John Wiley & Sons, 2012.
[11] Y. Sun, A. K. Wong, M. S. Kamel, "Classification of imbalanced data: a review", *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 23, no. 04, pp. 687-719, 2009.
[12] V. López, A. Fernández, S. García, V. Palade, F. Herrera, "An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics", *Information Sciences*, vol. 250, pp. 113-141, 2013.
[13] B. Frénay, M. Verleysen, "Classification in the presence of label noise: a survey", *Neural Networks and Learning Systems IEEE Transactions on*, vol. 25, no. 5, pp. 845-869, 2014.
[14] I. Cohen, F. G. Cozman, N. Sebe, M. C. Cirelo, T. S. Huang, "Semi supervised learning of classifiers: Theory algorithms and their application to human-computer interaction", *Pattern Analysis and Machine Intelligence IEEE Transactions on*, vol. 26, no. 12, pp. 1553-1566, 2004.
[15] Q. Yan, FR. Yu, Q. Gong, J. Li, "Software-Defined Networking (SDN) and Distributed Denial of Service (DDoS) Attacks in Cloud Computing Environments: A Survey Some Research Issues and Challenges", *IEEE Communications Surveys and Tutorials*, vol. 18, no. 1, pp. 602-622, 2016.
[16] T. Chen, C. Guestrin, "Xgboost: A Scalable Tree Boosting System" in KDD, pp. 785-794, 2016.