UTILISING MACHINE LEARNING TO PREVENT PHISHING URLS IN CYBER SECURITY

Alicja Margraf - 21123211

ABSTRACT

Cybercrime has become an increasing concern due to the development of technology, and the ability of hackers to steal personal information. Phishing URLs have especially been on the rise and using arising trends from post Covid-19. This study looks at addressing three issues in relation to phishing URLs: classification, clustering, and anomaly detection. Supervised learning was used for classification where it revealed perfect scores for ensemble methods and very high scores for non-ensemble methods such as GNB (Gaussian Naïve Bayes) and Decision Tree. XGBoost was chosen as the best performing due to its efficient computational time. For clustering, unsupervised algorithms such as K-Means++, DEC (Deep Embedded Clustering), hierarchical clustering, and DBSCAN were applied. DEC and hierarchical clustering achieved high silhouette scores, making clear clusters. Despite this, they achieved a low ARI showing the clusters were not representative of true labels. K-Means++ had the most balanced results making it the chosen algorithm to solve this problem. Lastly, to detect anomalies VAE (Variational Autoencoders) and IF (Isolation Forest) was used to identify threats whilst also utilising word embedding techniques to improve their performance. The result showed a significant improvement when word embedding was used and VAE with Word2Vec was the chosen algorithm for this problem. The original dataset was large, therefore computationally expensive and required more memory storage. This research manipulated the data to be more efficient to work with by balancing the dataset and using feature selection. It presents the importance of computational efficiency alongside model performance. The findings provide valuable insights to combat phishing through machine learning and improve security systems.

TABLE OF CONTENTS

INTRODUCTION

Phishing attacks, a form of social engineering where attackers deceive people to reveal sensitive information, have become more sophisticated due to new techniques emerging (Prasad & Chandra, 2024). This report acknowledges the risk of these threats and applies data on phishing URLs (Uniform Resource Locators) into machine learning algorithms to reveal most efficient ways to improve security. Exploratory Data Analysis is used to understand the data. Then it tackles the aim of this report and addresses three challenges: classification, clustering, and anomaly detection.

DOMAIN DESCRIPTION

Continuous technology advancement increases the risk of cybercrime. The estimated cost of cybercrime worldwide is $9.22 trillion in 2024, projecting to $15.63 trillion in 2029 (Petrosvan, 2024). Verizon revealed that 36% of breaching cases involve phishing (Ansari et al., 2022). Since Covid-19, it was discovered that phishing web attacks have increased significantly in contrast to email attacks, as shown in Fig. 1.

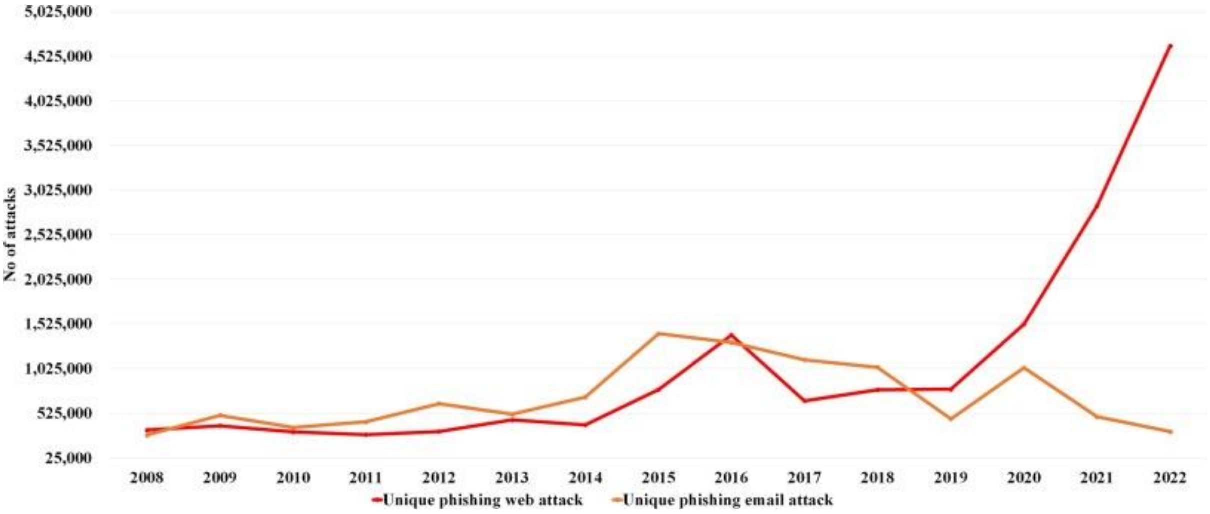


Fig. 1: Phishing web and email attacks (Prasad & Chandra, 2024)

These attacks can lead to financial loss, identity theft, reputation damage and malware. Interviews conducted by the Office for National Statistics reported that majority of fraudsters, in 2022, impersonated delivery companies, banks, and ecommerce companies (refer to Fig. 2). This emphasises attackers are leveraging behavioural online habits of post Covid-19. Alongside, they revealed individuals aged 25 to 44 are most likely targeted in the UK (Office for National Statistics, 2022).

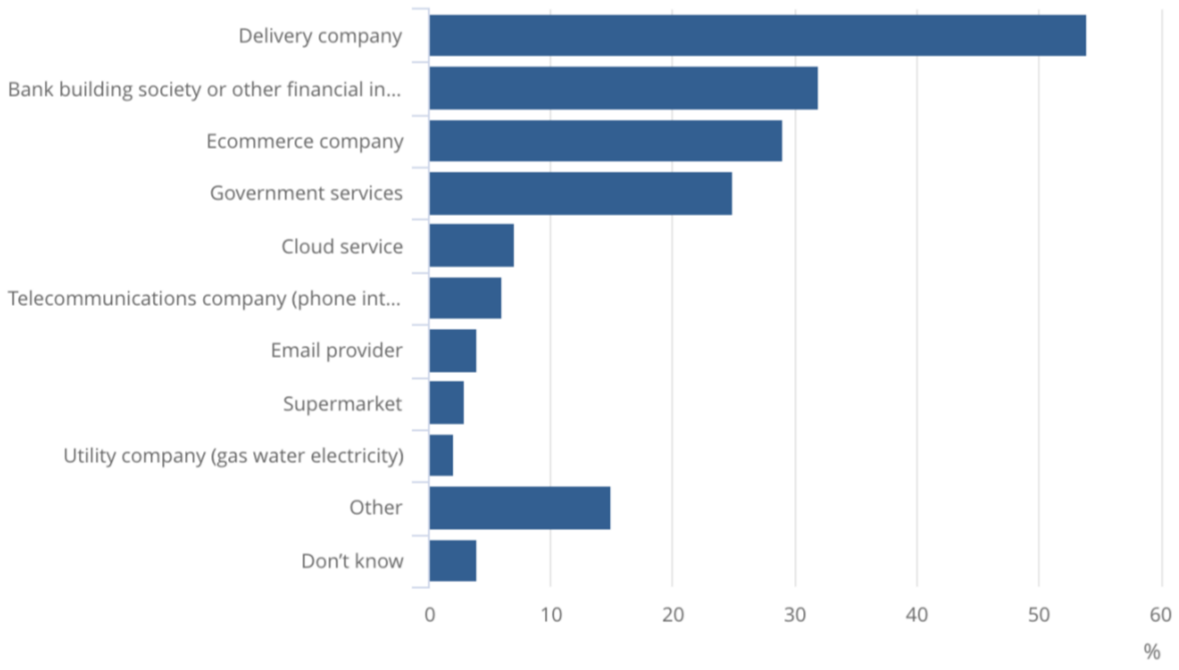


Fig. 2: Fraudster's choice of companies (Office for National Statistics, 2022)

Many features of a URL can be analysed to spot phishing like domain names, https and spelling mistakes (Gov, n.d.). Trying to tackle these manually would be timely and difficult, hence machine leaning algorithms are useful. Machine learning has proven to be successful through systems such as PhishWHO achieving 96.10% accuracy (Korkomaz et al., 2020).

Problem Definition

The goal of this report is to explore most effective machine learning algorithms to combat the increasing risk of phishing URLs. The following problems will be investigated:

Problem 1

This problem looks at phishing URL classification. The foundation of a classifying fraud detection process is a decision tree (Chy & Buadi, 2024). This report will use various supervised ensemble and non-ensemble algorithms. A range of metrics like accuracy, recall, precision, f1 score, and training time will be implemented to gain a deeper understanding. The objective is to create a model that accurately classifies URLs, whilst being computationally efficient, which will allow for a quicker removal of these to protect users.

Problem 2

This problem involves clustering URLs to identify patterns and find borderline cases. This uses an unsupervised approach. The outcome will show a visualisation of patterns which can assist in understanding high risk URLs that can pass undetected. The objective is that the model will create distinct groups that match the true label without using predefined labels.

Problem 3

This problem involves finding anomalies in an efficient way for further inspection. Doing this can potentially uncover unseen patterns and improve security systems to adapt to evolving attacks. This will also use an unsupervised approach with an outcome of a visualisation. The objective is to find clear anomalies that use unusual techniques.

Data Description

The dataset was extracted from UC Irvine Machine Learning Repository. It is comprised of 235,795 rows and 56 columns; it is a CSV file. There are 134,850 legitimate and 100,945 phishing URLs.

This data is substantial. This leads to positives like a variety of data trends and choice in relation to feature extraction. Nonetheless, it is more time consuming as models will require more processing power and memory. Additionally, feature extraction becomes more difficult to implement correctly.

This dataset has features such as URL length, letter ratio in URL and more. URLs containing between 54 to 75 characters can be labelled as suspicious and above 75 are most likely phishing (Mohammad et al., 2015). Fig. 3 shows the probability of characters in phishing and legitimate URLs, showing numbers and letters like ‘x’ are more likely in phishing.

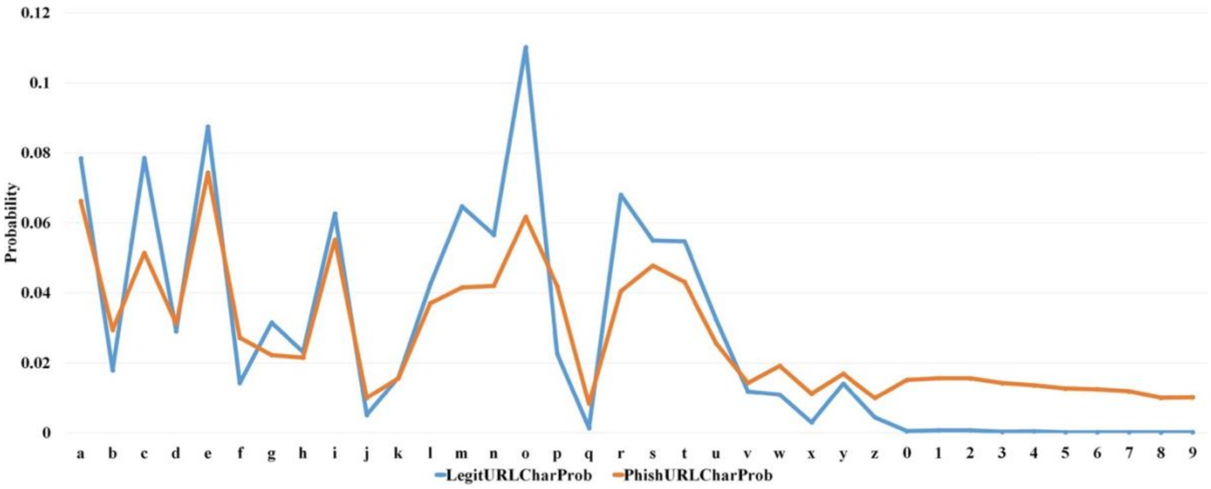


Fig. 3: Character probability (Prasad & Chandra, 2024)

The data has information on domain features such as subdomains, domain length and more. Attackers usually use multiple subdomains as they can include familiar company names. Furthermore, prefixes such as ‘- ‘are usually not used in legitimate URLs (Kara et al., 2022). A limitation is no feature on registration length or when it was created. This would be valuable, as domains aged over six months and not expiring within a year or less are likely legitimate (Mohammad et al., 2015).

There is information on TLDs (Top-Level Domains), these are the endings of a domain. Fake TLDs are often used and new generic TLDs have emerged such as ‘.zip’ (Ullrich, 2023). This can be problematic as cyber security tools can struggle to recognise them. This dataset includes TLD, TLD length, and TLD legitimate probability.

An Excel spreadsheet was created with feature descriptions, ranges, averages, and modes. The mean provides a summarising number for each column, the range shows its diversity, and mode displays most common categories.

ITEM	LINK	COMMENT
DATASET	https://archive.ics.uci.edu/dataset/967/phiusiil+phishing+url+dataset	The dataset size is 54.2MB
DESCRIPTIONS AND CALCULATIONS	https://mailbcuac-my.sharepoint.com/:x:/g/personal/alicja_margraf_mail_bcu_ac_uk/EfSO6nDJbJhDh7eHWl2DefEByezyH9YHhu4xZKR5Cuwi4g?e=7vF8kt	Excel file, one drive link shared to anyone with a BCU login. If link does not work refer to Appendix 1.

Exploratory Data Analysis

EDA consist of familiarising yourself with the data and using visual inspection (Cox, 2017). Due to the size of the data this step is essential. Univariate, bivariate, and multivariate EDA will be implemented, this includes using a different number of variables for analysis (Komorowski et al., 2016). The data does not include any missing values and duplicates which improves the reliability of the visualisations.

Univariate

This section looks at one column at a time. Fig. 4 firstly shows the datatypes in the data which revealed that the data is integer heavy, with 41 columns containing integers.

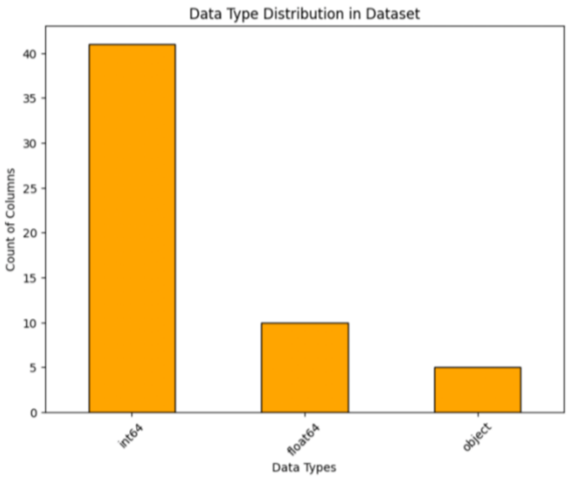


Fig. 4: Data types bar chart

TLD

Fig. 5 shows the most common TLD is '.com'. TLDs with three characters are the most popular. 'Other' in this figure represents TLDs worth 1% or less showing the versatility of them. Fig. 6 highlights two spikes, one close to 0 indicating suspicious TLDs, and another around 0.5 suggesting a neutral TLD which might require context.

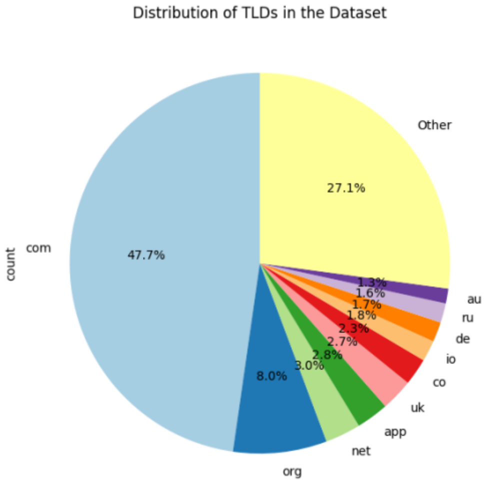


Fig. 5: Distribution of TLDs

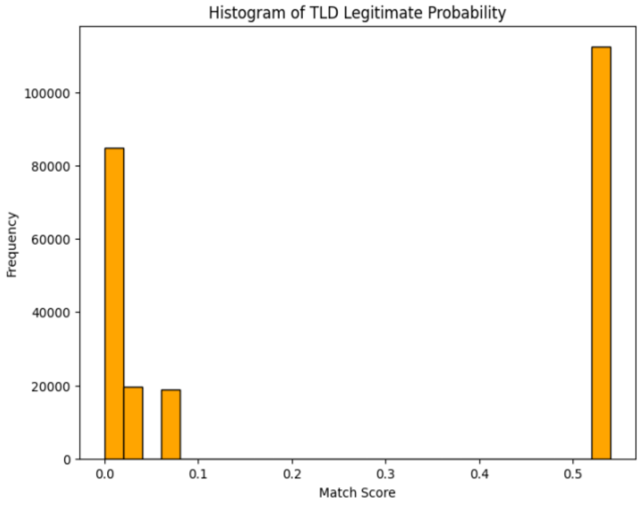
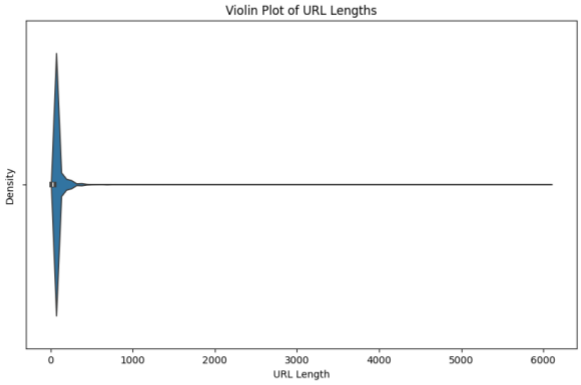


Fig. 6: Distribution of TLD Legitimate Probability

URL Length

Appendix 1 shows the average URL length is 35 characters however the range reaches 6097. This is an indicator of outliers. Attackers usually use longer URLs to hide the links true destination (Ojewumi et al., 2022). This shows the feature is right skewed, inferring the malicious URLs are increasing the range. The violin plot below (Fig, 7 & 8) shows density, there are very few data points past 300 characters as the line becomes flat. Most URLs have less than 100 characters, suggesting most phishing URLs try to hide themselves amongst legitimate URLs.



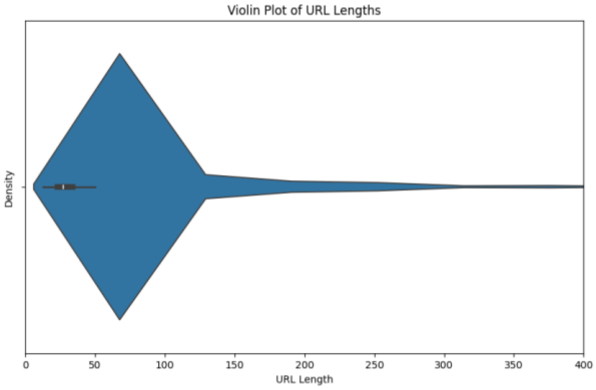


Fig. 7: URL Length Distribution

Fig. 8: Fig. 7 Zoomed in

Line Of Code

The line of code shows the websites complexity. Fig 9 shows a wide variety in the dataset, the range starts from 2 to 442,666. The most frequent is 2 to 100 lines of code showing popularity of less complex websites. This feature can be valuable to the algorithms as the wide spread of data will allow them to learn more effectively. Therefore, improve their ability to detect phishing website from the complexity of code.

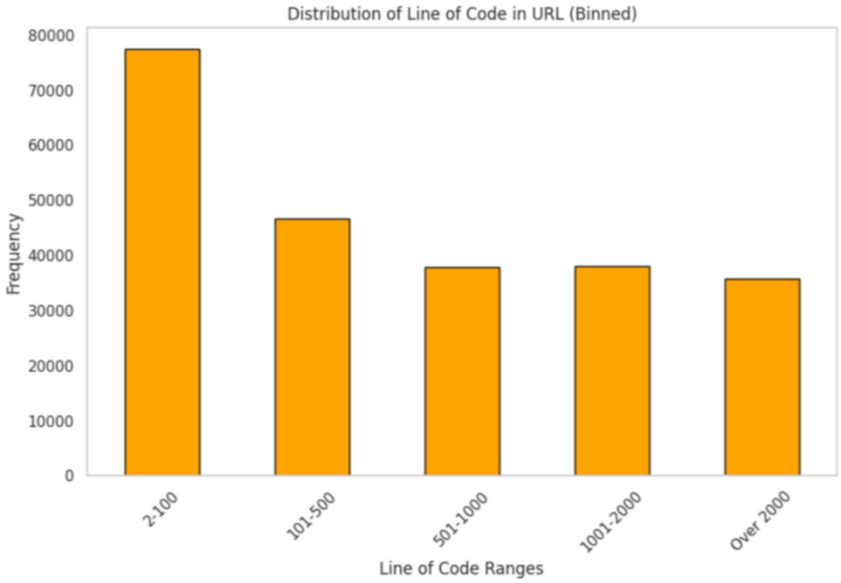


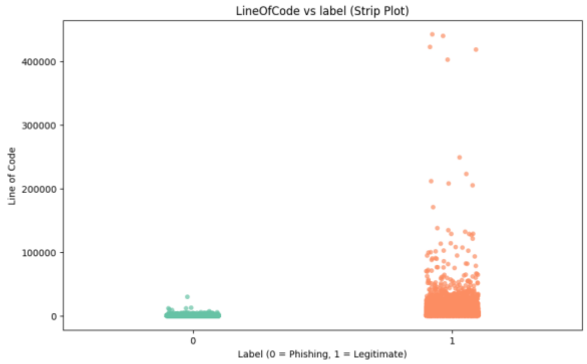
Fig. 9: Line of Code Distribution

Bivariate

This section helps to understand relationships between two features and is one of the simplest forms of statistical analysis (Sandhya et al., n.d.). It mainly concentrates on the relationship between the target and features.

Line of Code vs Label

Fig. 10 and 11 shows that phishing websites use substantially less lines of code. Even though there are instances with legitimate websites using a low number, a significant amount has over 2000. This figure emphasises the findings of phishing websites being less complex due to having a shorter duration of existence.



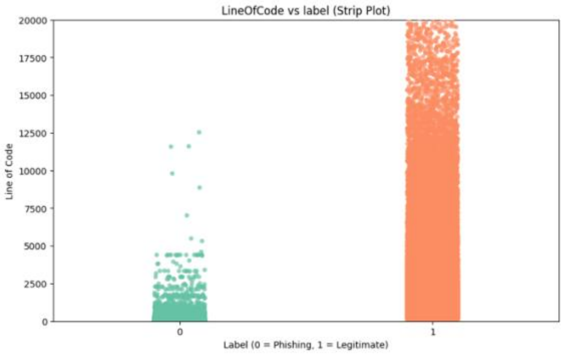


Fig. 10: Line of Code vs Label

Fig. 11: Fig. 10 Zoomed in

URLs vs Label

It can be inferred from the dataset that legitimate URLs are no longer than 60 characters (refer to Fig. 12 & 13). Most are approximately 30 characters long. On the other hand, phishing URLs go up to 6097 characters long. Therefore, any URL in this dataset with more than 60 characters can be classified as phishing, showing this features importance. Additionally, Fig. 14 shows that all legitimate URLs scored 100 on the URL similarity index. Meaning this feature would help with classification.

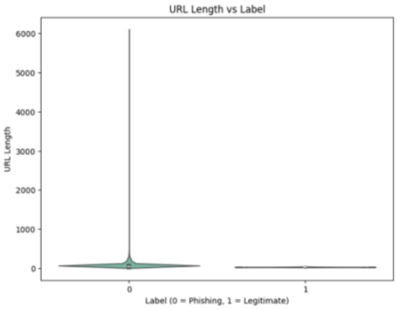


Fig. 12: URL Length vs Label.

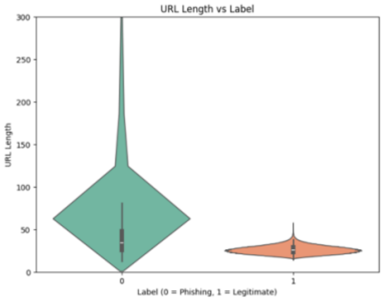


Fig. 13: Fig. 12 Zoomed in.

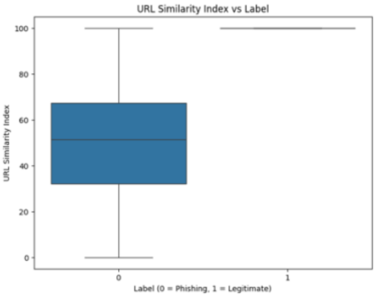


Fig. 14: URL Similarity Index vs Label

Social Networking vs Label

Social networking relates to websites including social media platforms and ways to communicate. Fig. 15 shows that nearly none of the phishing websites have social networking. This is likely because they are created for short-term purpose. It could increase the likelihood of detection and it would be timely to maintain legitimate looking social media accounts.

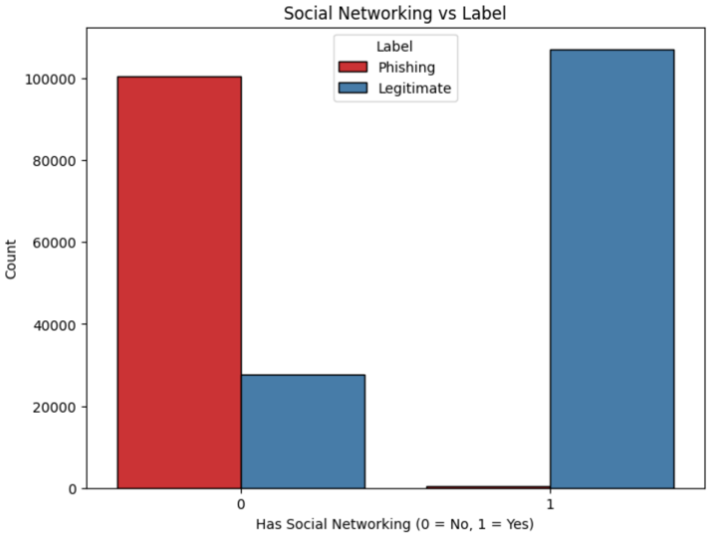


Fig. 15: Social Networking vs Label

Heatmap:

Appendices 2 and 3 show a heatmap with feature correlations. Dark red represents positive and blue shows negative correlations. The highly correlated columns have similar information and can be dropped to improve speed of models and reduce the probability of overfitting. These features can be removed:

- ‘LetterRatioInURL’: Highly correlated with ‘NoOfLettersInURL’
- ‘DegitRatioInURL’: Highly correlated with ‘NoOfDegitsInURL’

- ‘DomainLength’: Highly correlated with ‘URLLength’
- ‘NoOfObfuscatedChar’: Highly correlated with ‘ObfuscationRatio’
- ‘NoOfSelfRedirect’: Highly correlated with ‘NoOfURLRedirects’
- ‘DomainTitleMatchScore’: Highly correlated with ‘URLTitleMatchScore’

Multivariate

This section analyses multiple variables at once (Oluleye, 2023). PCA (Principal Components Analysis) is a technique that visualises multiple variables, however as it is linear and sensitive to outliers it did not do well with this data set. Instead as 2D t-SNE (t-Distributed Stochastic Neighbor Embedding) was created. This technique plots each row of data onto the graph and preserves local and global structures. However, it is computationally expensive as it took 1 hour 30 minutes to generate.

Fig. 16 reveals two distinct clusters, meaning the dataset is effective in grouping URLs. The purple area represents phishing and yellow legitimate. The URLs plotted near each other have similar characteristics. The outliers can be spotted easily. The phishing URLs appearing in the yellow section, especially far from the border, are more sophisticated.

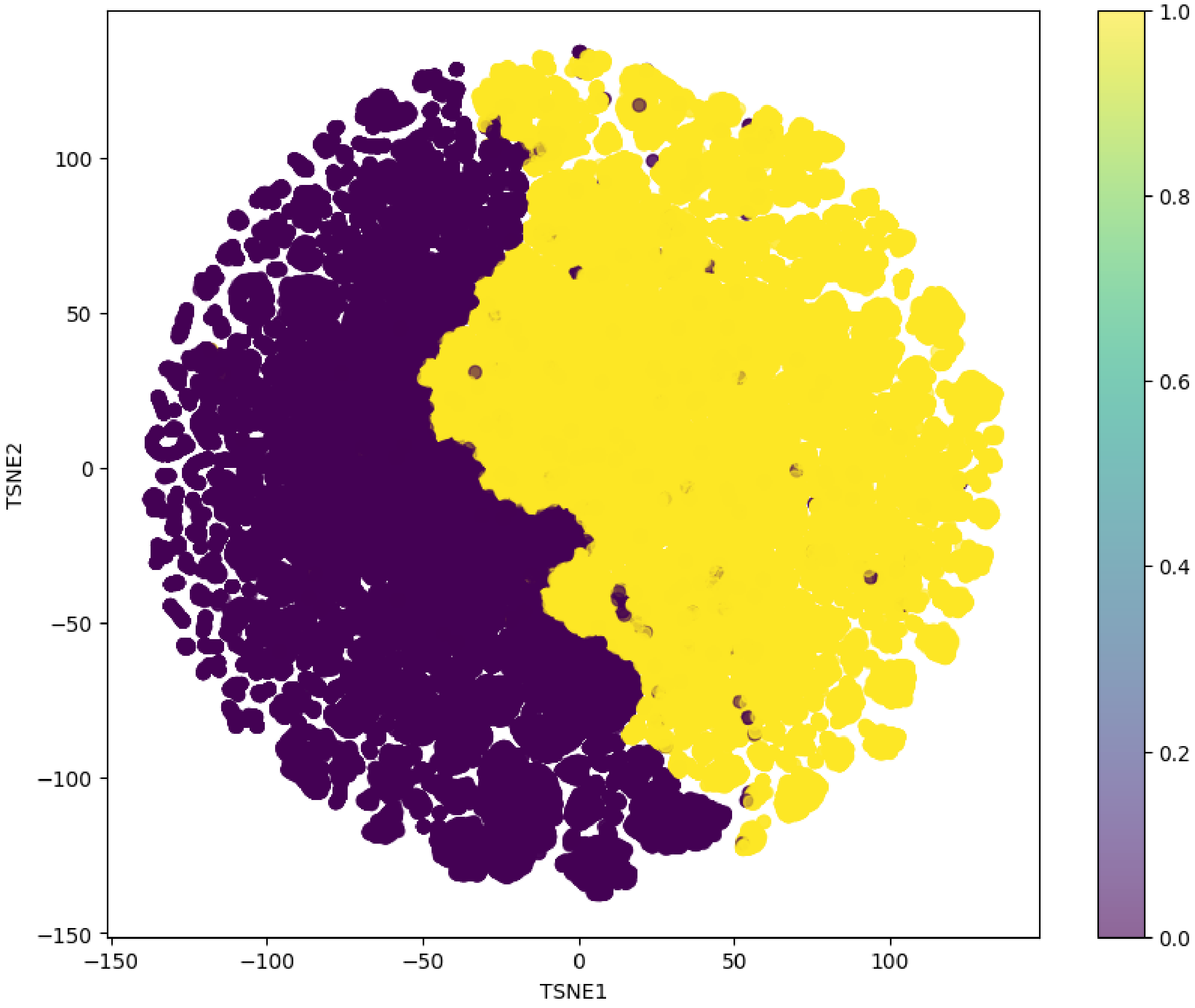


Fig. 16: t-SNE of entire dataset

Data Preparation and Optimisation:

Balancing The Dataset

The dataset consists of 42.8% phishing and 57.2% legitimate URLs which is quite balanced. Ideally, to avoid bias a 50% split would be better. As the dataset is already large an over-sampling technique like SMOTE would be inefficient. An under-sampling approach was taken instead. It reduces the majority to be equal to the minority (He & Garcia, 2009). This reduced the data to 201,890 rows. Loss of data was considered however each class still holds a significant number of examples.

Feature Removal

The features which were specified as highly correlated were dropped. Also, the 'FILENAME' column which provides no relevant information also got removed. These do not provide new insights for the models but provide overlapping information. This increases complexity and increases the probability of overfitting.

Alongside, the Recursive Feature Elimination (RFE) technique was used to identify and remove unimportant features. This technique uses an estimator to train on all features and evaluates their importance (Scikit, n.d.). This leaves the dataset with 42 columns.

Minimising The Dataset

It was challenging to determine whether the data should be minimised because of the large loss of data. Due to the potential risk of code crashes associated with the large dataset, 10% of each class was kept. The dataset now contains 20,188 rows. Another t-SNE was created to visualise the spread of the data to ensure the patterns were not lost (refer to Fig. 17). This revealed that clusters were preserved; therefore, this approach was taken to improve efficiency.

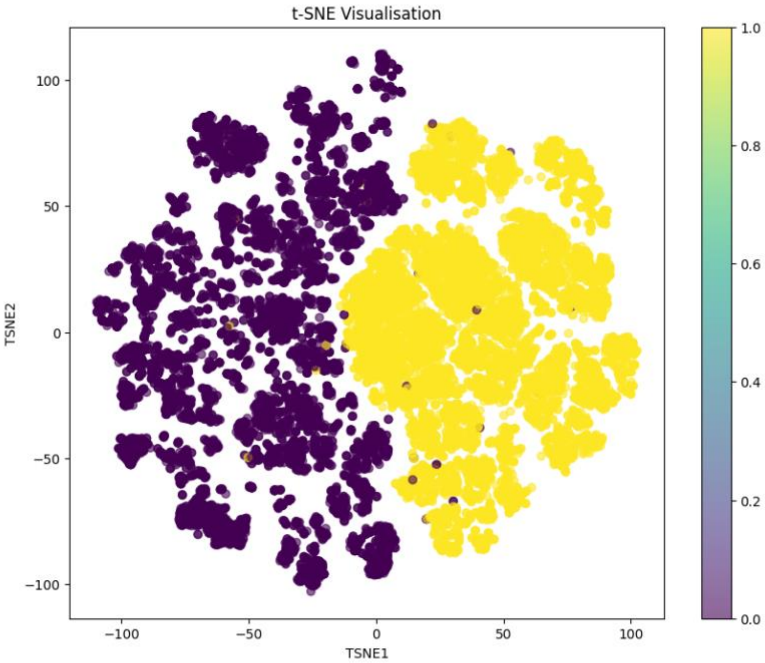


Fig. 17: t-SNE of minimised dataset

Label Encoding

A copy of the dataset was made using the label encoder because many machine learning algorithms require a numerical input. One hot encoding was the preferred choice as the strings in the dataset are nominal however memory usage was a problem. Label encoding was chosen because of its simplicity.

Problem 1 (Supervised): Classification

Research

Korkmaz et al., 2020:

The study conducted by Korkmaz used three datasets which can be found below to classify URLs.

	DATASET 1	**DATASET 2**	**DATASET 3**
PHISHING	40,668	40,668	40,668
LEGITIMATE	43,189	42,220	85,409
TOTAL	83,857	82,888	126,077

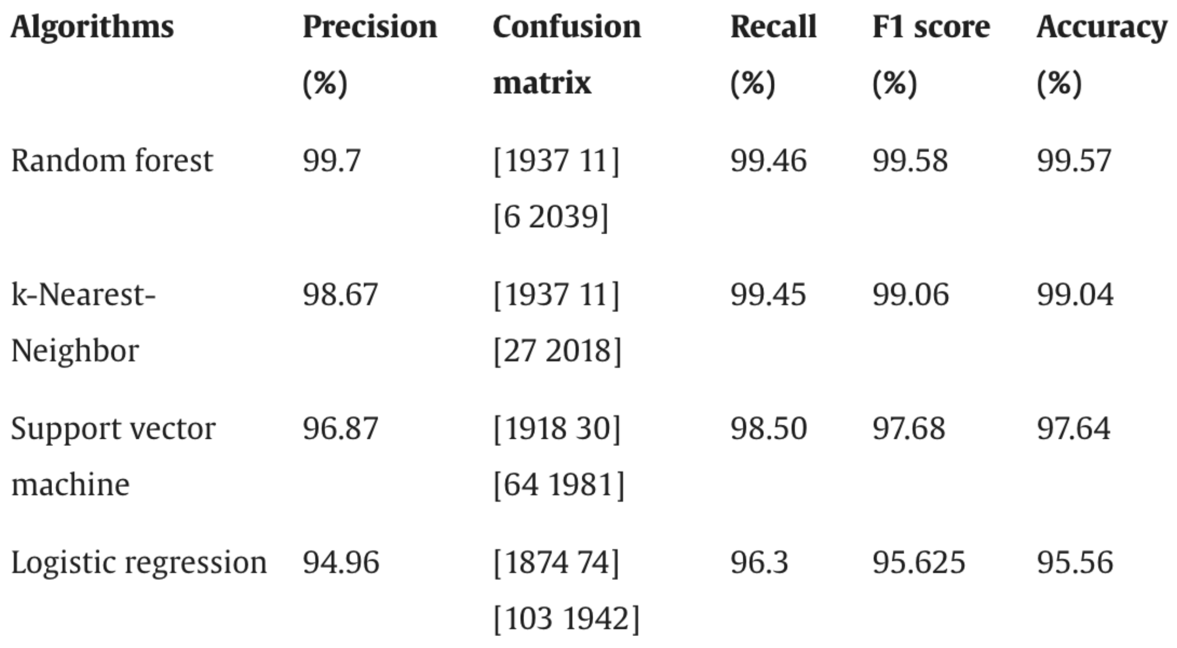
There were 48 features used after sorting them with a Random Forest Classifier. The result can be found below:

CLASSIFIER	**DATASET 1**		**DATASET 2**		**DATASET 3**	
	Accuracy (%)	**Time (sec)**	**Accuracy (%)**	**Time (sec)**	**Accuracy (%)**	**Time (sec)**
XGBOOST	92.95	622.6	83.69	395.6	83.27	271.4
RANDOM FOREST	**94.59**	784.3	**90.5**	439.1	**91.26**	133.2
LOGISTIC REGRESSION	91.31	20.5	75.65	51.7	78.26	63.5
K-NEAREST NEIGHBOR	91.49	413.7	81.47	154.5	81.11	262.1
SUPPORT VECTOR MACHINE	87.03	7537.4	70.2	9833.6	76.76	15956.2
DECISION TREE	92.59	18.8	81.67	23	81.66	17.8
ARTIFICIAL NEURAL NETWORK	94.35	57.6	88.22	40.3	88.88	29.4
NAIVE BAYES	88.35	3.9	70.05	1.8	67.04	2.1

The RF performed the most accurately however required more computational time in contrast to other models. The SVM model performed poorly considering the low accuracy and training time. This can be because of the model's nature of solving quadratic optimisation problems as it aims to calculate the best margin between data points for different classes. In conclusion ensemble methods performed well, demonstrating that older well – known algorithms can deliver strong results. The limitation to this research is it only evaluates time and accuracy without other metrics.

Gupta et al., 2021:

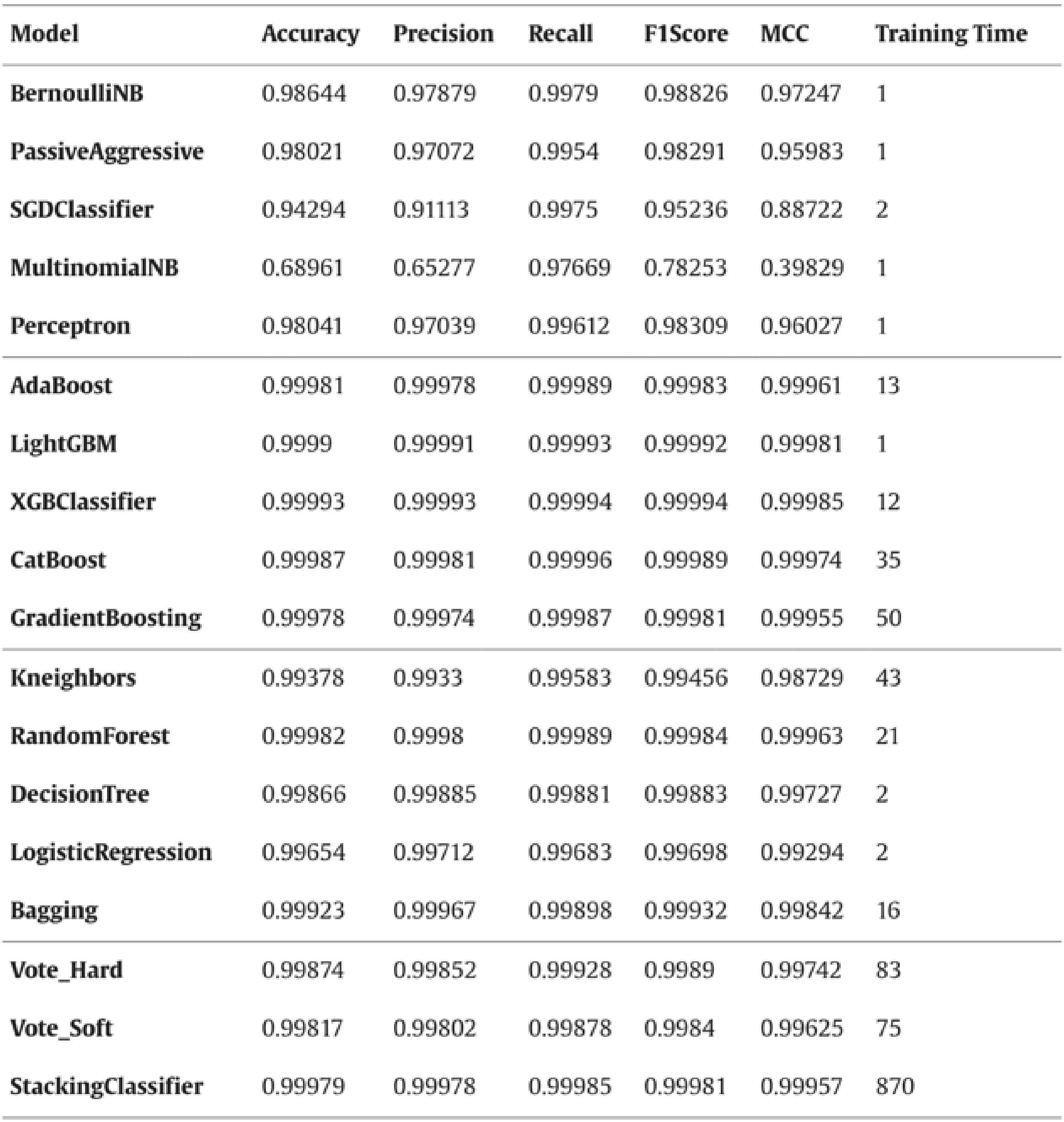
This research paper provided more metrics. RF also scores highest in all metrics in this paper, only misclassifying 17 cases. False negatives are most dangerous in this case as these are the phishing URLs which were classified as legitimate. Missing these can lead to undetected threats which leaves users vulnerable to data theft. RF had a false negative rate of 0.3%, which emphasises that ensemble methods should be concentrated on to solve this problem.



Algorithms	Precision (%)	Confusion matrix	Recall (%)	F1 score (%)	Accuracy (%)
Random forest	99.7	[1937 11] [6 2039]	99.46	99.58	99.57
k-Nearest-Neighbor	98.67	[1937 11] [27 2018]	99.45	99.06	99.04
Support vector machine	96.87	[1918 30] [64 1981]	98.50	97.68	97.64
Logistic regression	94.96	[1874 74] [103 1942]	96.3	95.625	95.56

Prasad & Chandra, 2024:

This research paper uses the same dataset containing all 235,795 instances. This research uses a wider variety of machine learning algorithms. It can be found that the top performers are different ensemble methods such as AdaBoost, LightGBM and XGBClassifier which are boosting methods. MultinomialNB was the worst in performance, even though it was quick to train this can be because of data complexity. Stacking Classifier also came out with good results however its complexity leads to high training time (870 seconds) which means it is not as practical. Other models match its performance with better training times meaning this model should not be prioritised.

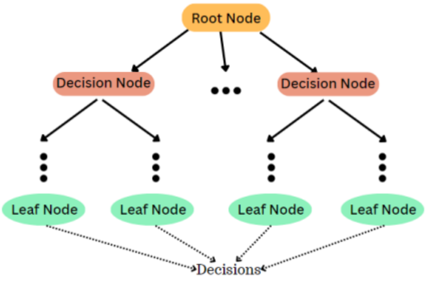


Model	Accuracy	Precision	Recall	F1Score	MCC	Training Time
BernoulliNB	0.98644	0.97879	0.9979	0.98826	0.97247	1
PassiveAggressive	0.98021	0.97072	0.9954	0.98291	0.95983	1
SGDClassifier	0.94294	0.91113	0.9975	0.95236	0.88722	2
MultinomialNB	0.68961	0.65277	0.97669	0.78253	0.39829	1
Perceptron	0.98041	0.97039	0.99612	0.98309	0.96027	1
AdaBoost	0.99981	0.99978	0.99989	0.99983	0.99961	13
LightGBM	0.9999	0.99991	0.99993	0.99992	0.99981	1
XGBClassifier	0.99993	0.99993	0.99994	0.99994	0.99985	12
CatBoost	0.99987	0.99981	0.99996	0.99989	0.99974	35
GradientBoosting	0.99978	0.99974	0.99987	0.99981	0.99955	50
Kneighbors	0.99378	0.9933	0.99583	0.99456	0.98729	43
RandomForest	0.99982	0.9998	0.99989	0.99984	0.99963	21
DecisionTree	0.99866	0.99885	0.99881	0.99883	0.99727	2
LogisticRegression	0.99654	0.99712	0.99683	0.99698	0.99294	2
Bagging	0.99923	0.99967	0.99898	0.99932	0.99842	16
Vote_Hard	0.99874	0.99852	0.99928	0.9989	0.99742	83
Vote_Soft	0.99817	0.99802	0.99878	0.9984	0.99625	75
StackingClassifier	0.99979	0.99978	0.99985	0.99981	0.99957	870

Machine Learning Algorithms

Decision Tree

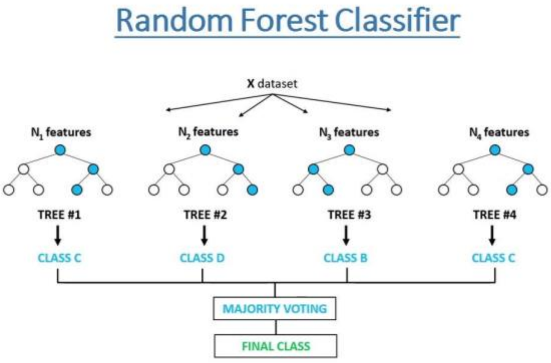
Decision trees were first implemented in the later decades of the 20th century. They split data into smaller groups based on input features. This creates nodes where data within each section is more similar, and groups are increasingly different (Ville, 2013).



(Gahlawat, 2023)

Random Forest (RF)

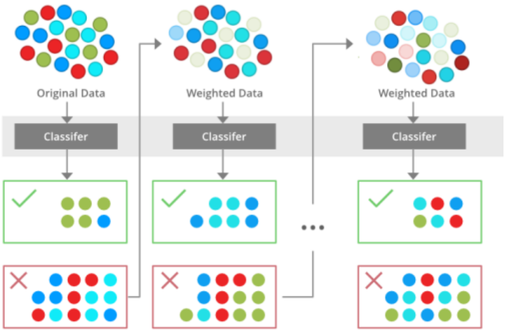
This method, developed by Breiman in 2001, is an ensemble learning algorithm; it uses the bagging sampling approach. It creates several decision tree classifiers on various sub-samples of the data. Majority voting is used to classify the instance (Fawagreh et al., 2014).



(Chauhan, 2021)

XGBoost

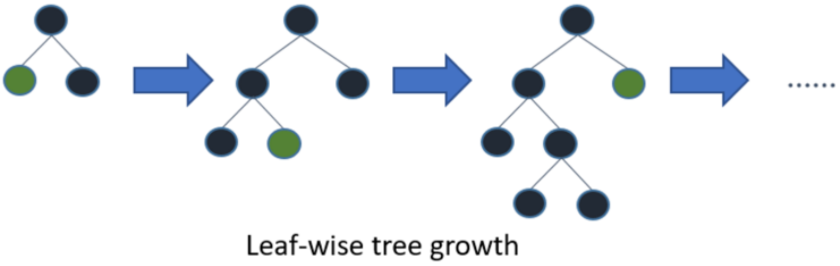
Extreme Gradient Boost is an ensemble learning method which combines decision trees with gradient boosting (Moore & Bell, 2022). It was developed by Tianqi Chen in 2014. Boosting trains each model based on the misclassification of the previous model which improves its accuracy (Torres, 2023).



(GeeksforGeeks, 2023)

Light GBM

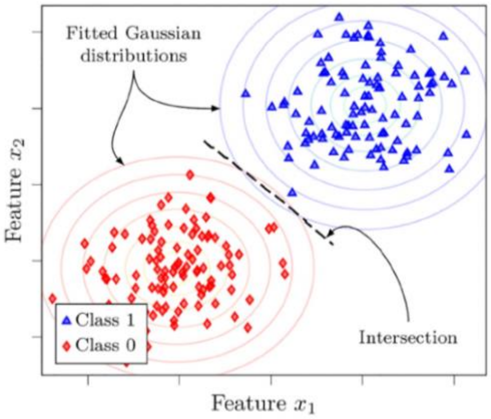
This is a tree- based ensemble learning method developed by researchers at Microsoft and Peking University (Soomro et al., 2024). This algorithm uses both gradient-based one side sampling and exclusive feature bundling. It can outperform XGBoost in terms of computational speed and memory consumption (Ke et al., n.d.).



(Mandot, 2017)

Gaussian Naïve Bayes (GNB)

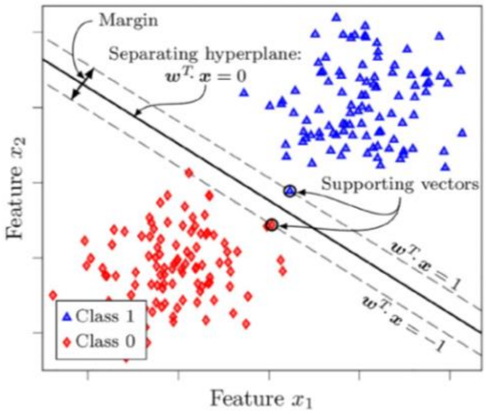
This is a simple algorithm which classifies data by finding the class with the highest probability, assuming features are independent and follow Gaussian distribution. It uses a formula which combines distance, variance, and prior probabilities (Ontivero-Ortega et al., 2017).



(Shyrokykh et al., 2023)

Support Vector Machine (SVM)

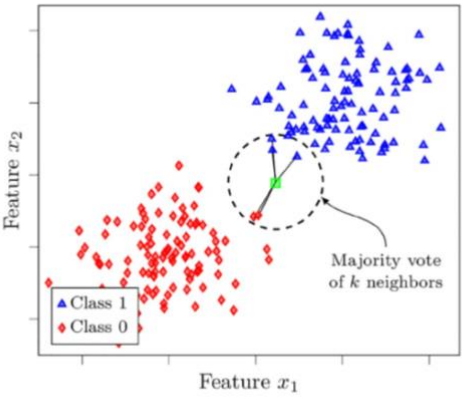
It was first introduced for linear problems and later generalised to non-linear. This algorithm fits a hyperplane that separates the classes and maximises the margin. Only support vectors impact the training which are the samples closest to the hyperplane (Shyrokykh et al., 2023).



(Shyrokykh et al., 2023)

k-Nearest Neighbor (kNN)

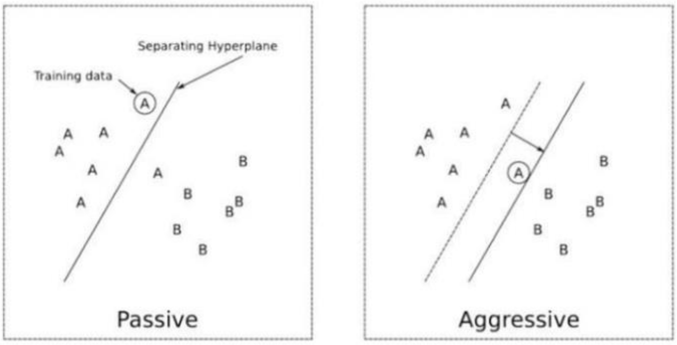
This algorithm classifies data points based on their distance to nearby points, without a training phase. It assigns a class to a new point by voting among the classes of its closest neighbours (Shyrokykh et al., 2023).



(Shyrokykh et al., 2023)

Passive Aggressive (PA)

It is an online-learning algorithm, and it is well suited to receive data in a continuous stream as it does not require a lot of memory. This algorithm uses passive when the prediction is correct which leads to no change in the model but uses aggressive if the prediction is not correct to change the model accordingly (Kumar, 2023).



(Kumar, 2023)

RESULTS:

MODEL	ACCURACY	PRECISION	RECALL	F1- SCORE	TRAINING TIME (SECONDS)	CONFUSION MATRIX LABEL 0 = PHISHING LABEL 1 = LEGITIMATE
XGBOOST	1.00	**Label 0:** 1.00 **Label 1:** 1.00	**Label 0:** 1.00 **Label 1:** 1.00	**Label 0:** 1.00 **Label 1:** 1.00	0.302	3025 0 0 3032
LIGHT GBM	1.00	**Label 0:** 1.00 **Label 1:** 1.00	**Label 0:** 1.00 **Label 1:** 1.00	**Label 0:** 1.00 **Label 1:** 1.00	0.433	3025 0 0 3032
RF	1.00	**Label 0:** 1.00 **Label 1:** 1.00	**Label 0:** 1.00 **Label 1:** 1.00	**Label 0:** 1.00 **Label 1:** 1.00	1.791	3025 0 0 3032
GNB	0.99	**Label 0:** 0.98 **Label 1:** 1.00	**Label 0:** 1.00 **Label 1:** 0.97	**Label 0:** 0.99 **Label 1:** 0.99	0.019	3024 1 77 2955
DT	0.99	**Label 0:** 0.98 **Label 1:** 1.00	**Label 0:** 1.00 **Label 1:** 0.97	**Label 0:** 0.99 **Label 1:** 0.99	0.093	3024 1 77 2955
KNN	0.99	**Label 0:** 1.00 **Label 1:** 0.99	**Label 0:** 0.99 **Label 1:** 1.00	**Label 0:** 0.99 **Label 1:** 0.99	125.246	2999 26 11 3021
PA	0.98	**Label 0:** 0.99 **Label 1:** 0.97	**Label 0:** 0.97 **Label 1:** 0.99	**Label 0:** 0.98 **Label 1:** 0.98	0.218	2945 80 43 2989
SVM	0.97	**Label 0:** 1.00 **Label 1:** 0.94	**Label 0:** 0.94 **Label 1:** 1.00	**Label 0:** 0.97 **Label 1:** 0.97	3.424	2843 182 9 3023

As expected, all the machine learning algorithms showed high accuracy results for classifying URLs. The table is organised based on these accuracies. XGB, Light GBM, and RF all scored the same on all metrics except training time with XGB being most efficient and accurate. From previous research, especially Prasad and Chandra's which uses the original dataset, these perfect scores are not worrying as the data set was further balanced and noise was removed; this could have led to the slight improvement in scores. Fig. 18 shows the RF feature importance. The URL Similarity Index and line of code scored the highest. The least important features were dropped for RF which slightly improved training time.

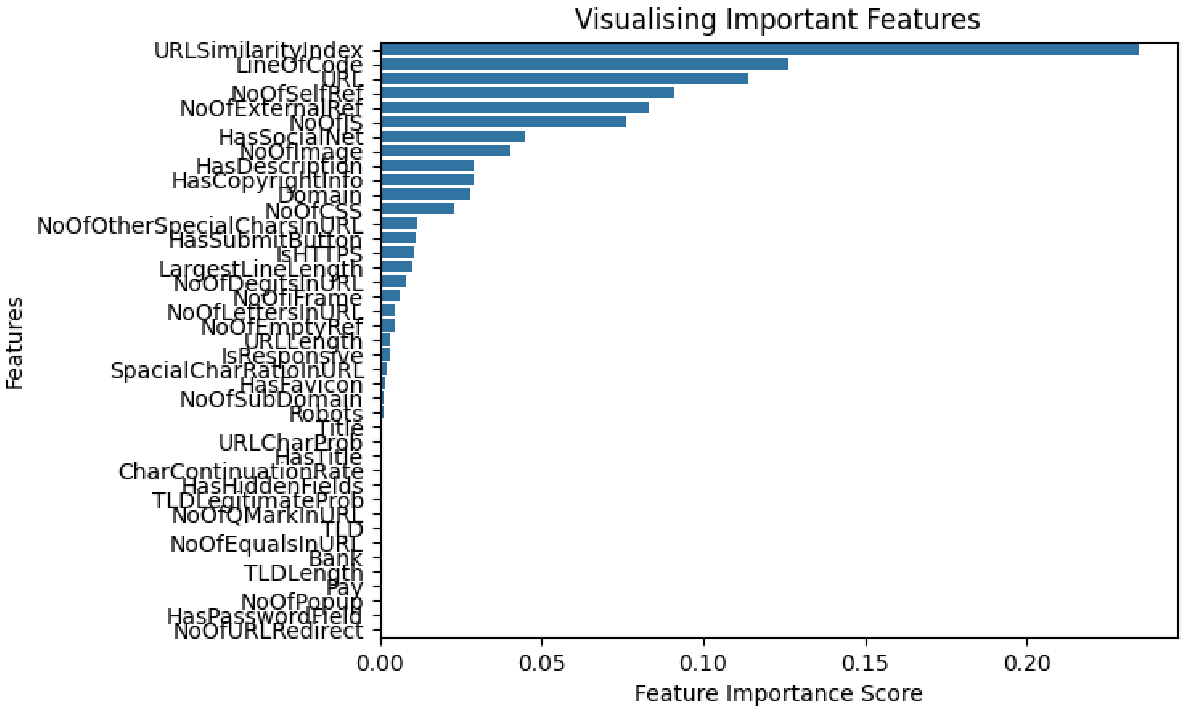


Fig. 18: RF Feature Importance

Non-ensemble methods such as GNB and DT also performed well with slight imperfections in recall and precision both only misclassifying 78 samples (refer to Fig. 19). When looking at the confusion matrix they only misclassified one phishing URL (refer to Fig. 20) and both had quicker training times than the ensemble methods. Therefore, these can be argued as better to use in certain circumstances if computational efficiency is a priority. In real- time application their simpler nature would be useful to make decisions instantly however due to the occasional misclassification additional safeguarding checks could be implemented.

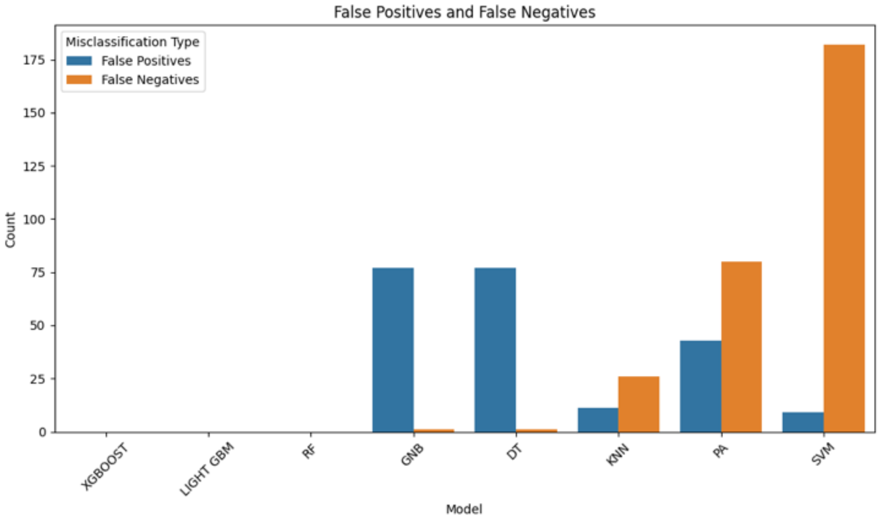


Fig. 19: Problem 1 Confusion Matrix Findings

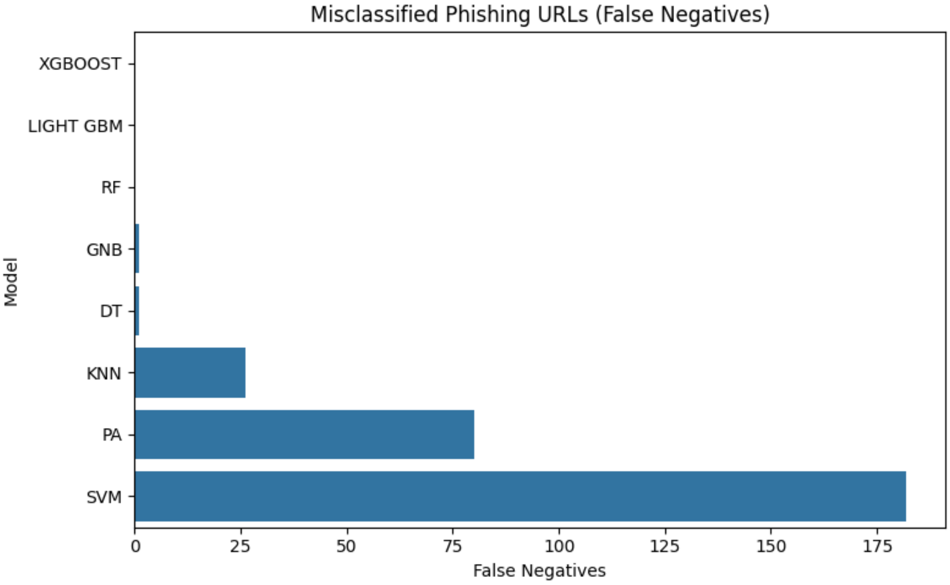


Fig. 20: Problem 1 False Negatives

The high training time of kNN and SVM would be impractical to implement. Furthermore, they have misclassified more phishing URLs which is a significant risk. Even though, their scores are still high they would be less suitable for solving this problem where speed and reliability is critical.

Problem 2 (Unsupervised): Clustering

Research:

Radi & Moughit, 2023:

This paper focuses on implementing k-means (the most popular clustering approach), DBSCAN, hierarchal clustering and Gaussian mixture model to detect malicious activity in DNS (Domain Name Systems). To further improve performance, it implements PCA and t-SNE. The research found that K-means and hierarchal clustering was the most accurate and MMG performed the worst. This research paper only used similarity score as a metric which is a limitation, Fig. 21 shows the findings.

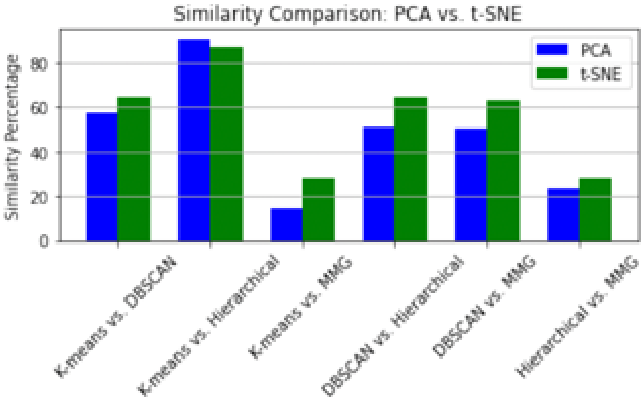


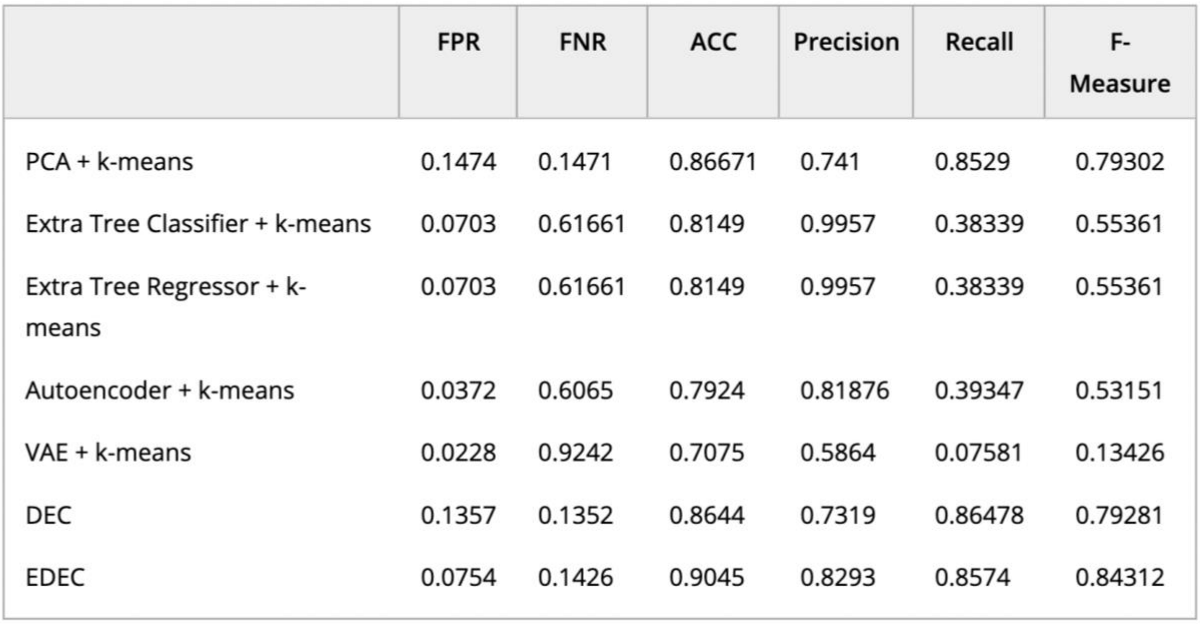
Fig. 21: Radi & Moughit, 2023

Xie et al., 2016:

This paper found that popular methods like k-means and GMM tend to be ineffective when input dimensionality is high. They proposed an algorithm called deep embedded clustering (DEC). They used 4 different datasets with different dimensions to test the algorithm. The results displayed that DEC was the most accurate for each.

Ng et al., 2019:

This paper looks at the implementation of DEC in cyber security and its effectiveness. It states that clustering is proven to be one of the effective ways to analyse malwares. The results show that DEC produces the lowest false negative rate which is a priority to improve security. However, its false positive rate was higher than the other algorithms. This is not as important as unnecessary alerts do not pose significant threat compared to undetected malware. This paper also finds that Enhanced Deep Embedded Clustering outperforms DEC.



	FPR	FNR	ACC	Precision	Recall	F-Measure
PCA + k-means	0.1474	0.1471	0.86671	0.741	0.8529	0.79302
Extra Tree Classifier + k-means	0.0703	0.61661	0.8149	0.9957	0.38339	0.55361
Extra Tree Regressor + k-means	0.0703	0.61661	0.8149	0.9957	0.38339	0.55361
Autoencoder + k-means	0.0372	0.6065	0.7924	0.81876	0.39347	0.53151
VAE + k-means	0.0228	0.9242	0.7075	0.5864	0.07581	0.13426
DEC	0.1357	0.1352	0.8644	0.7319	0.86478	0.79281
EDEC	0.0754	0.1426	0.9045	0.8293	0.8574	0.84312

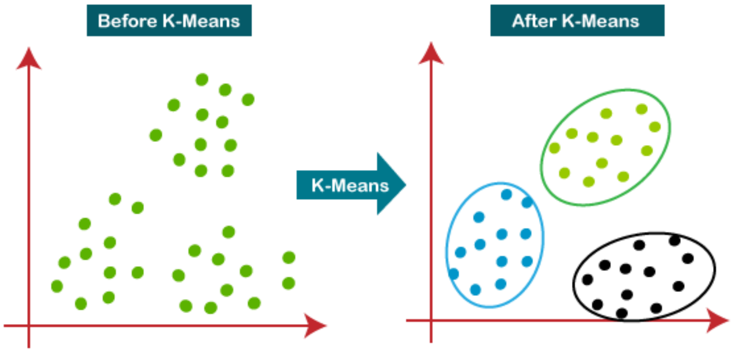
Feyereisl & Aickelin, 2009:

This paper looks at the Self-Organising Maps model and its application in cyber security. SOM is usually used for anomaly detection and has been proven effective by itself and in combination with other models. It can learn normal behaviour of a system to identify anomalies by organising clusters based on similarity.

Machine Learning Algorithms

K-Means

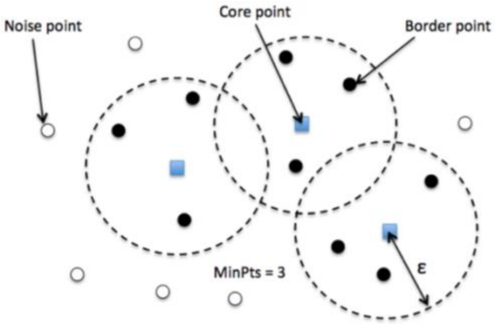
This is a centroid-based clustering algorithm that splits the data into K clusters, where K represents a chosen number. It assigns each data point to the nearest centroid based on Euclidean distance and recalculates the centroid as the mean of all data points assigned to that cluster (Radi & Moughit, 2023).



(Prateek, 2022)

DBSCAN

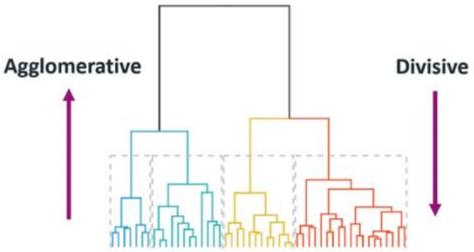
A density-based algorithm which begins with selecting a random data point, it will then create a cluster by adding points within Epsilon distance of this data point. It will become a core point if it has MinPts neighbours within that distance and its neighbours are added to the cluster. The points which do not meet the density requirement are considered as noise (Radi & Moughit, 2023).



(Chauhan, 2022)

Hierarchical Clustering

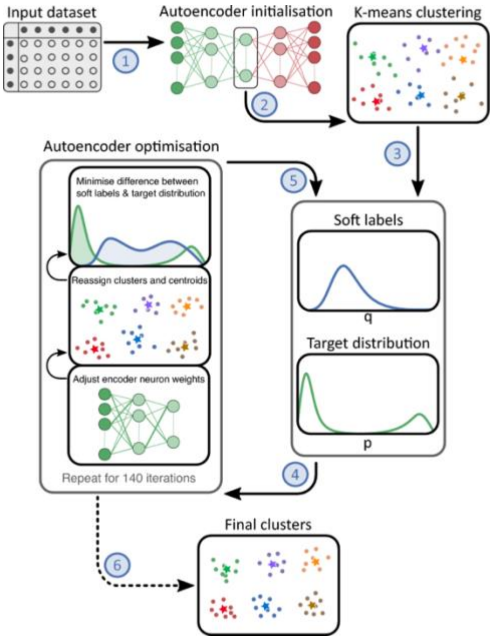
This algorithm builds a hierarchy of clusters. It considers each data point as a separate cluster and merges the closest clusters together based on chosen linkage criteria (Radi & Moughit, 2023).



(Re, 2023)

Deep Embedded Clustering (DEC)

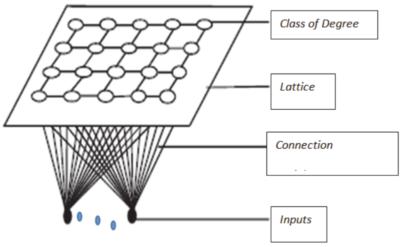
This algorithm was developed by Xie and others, it uses an autoencoder, which is a type of neural network that simplifies large data. This is then used to make clusters. Firstly, it uses K-means and then the loss function is incorporated to improve the autoencoder (Kok et al., 2024).


Input dataset
Autoencoder initialisation
K-means clustering
1
2
3
Autoencoder optimisation
5
Minimise difference between soft labels & target distribution
Reassign clusters and centroids
Adjust encoder neuron weights
Repeat for 140 iterations
Soft labels
q
Target distribution
p
4
6
Final clusters

(Kok et al., 2024)

Self-Organising Map (SOM)

This model was introduced in 1980 by Tuevo Kohonen it maps points on a two-dimensional grid based of similarity therefore it is perfect for classification (Bishayee, 2023).



(Inyang et al., 2024)

Results

MODEL	SILHOUETTE SCORE	ADJUSTED RAND INDEX (ARI)
HIERARCHICAL CLUSTERING	0.9745	0.0000
DEC	0.9707	0.0001
SOM	0.8851	0.0006
K-MEANS ++	0.2205	0.9635
DBSCAN	-0.2995	0.0294

Fig. 21 shows a visualisation of the results. The model which performed best in terms of silhouette score was hierarchical clustering, meaning the cluster quality was the best. DEC and SOM also had high silhouette scores. This suggests they are capturing hidden patterns and data relationships better. However, the ARI is low which means the clusters are not aligning with the true labels. On the other hand, K-Means ++ has the highest ARI score which means the algorithms is clustering based on true labels better however the clusters are not as clear. In the context of analysing borderline cases K-Means ++ would be the most suitable algorithm due to reasonably formed clusters, its computationally efficient nature, and it aligns well with true behaviours. The other algorithms are also very informative but for underlying data structures. DBSCAN performed poorly on both.

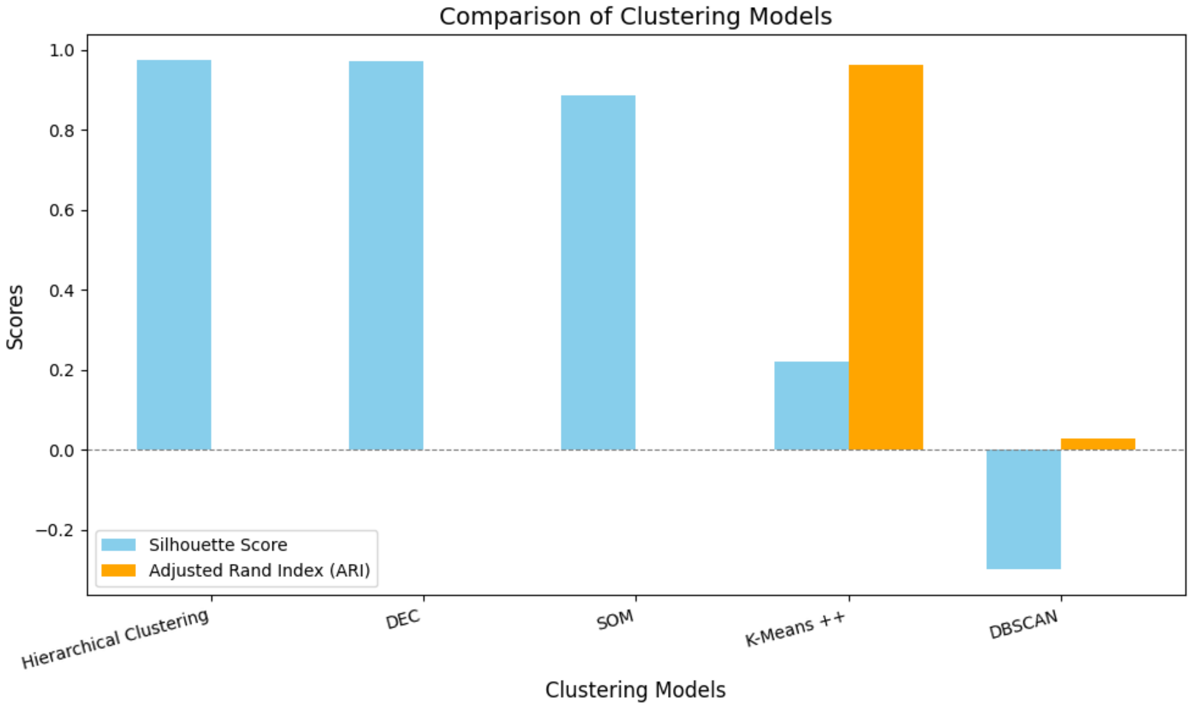


Fig. 21: Clustering results

K-MEANS ++

Fig.22 shows the elbow method, it presents that the best number for k is 3. However, as the elbow method can still be seen as unclear when running K-Means++ a loop was created to show from k=2 to k=5.

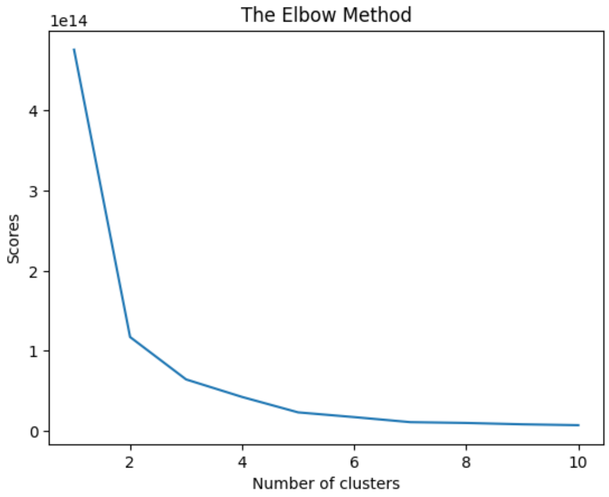


Fig.22: The Elbow Method

PCA was trialled for visualisation however since it was unclear, a t-SNE visual and scores on each value of k were created. This revealed that k=2 had the best scores with the highest ARI and similar Silhouette score to k = 3 which can be seen in Fig. 23.

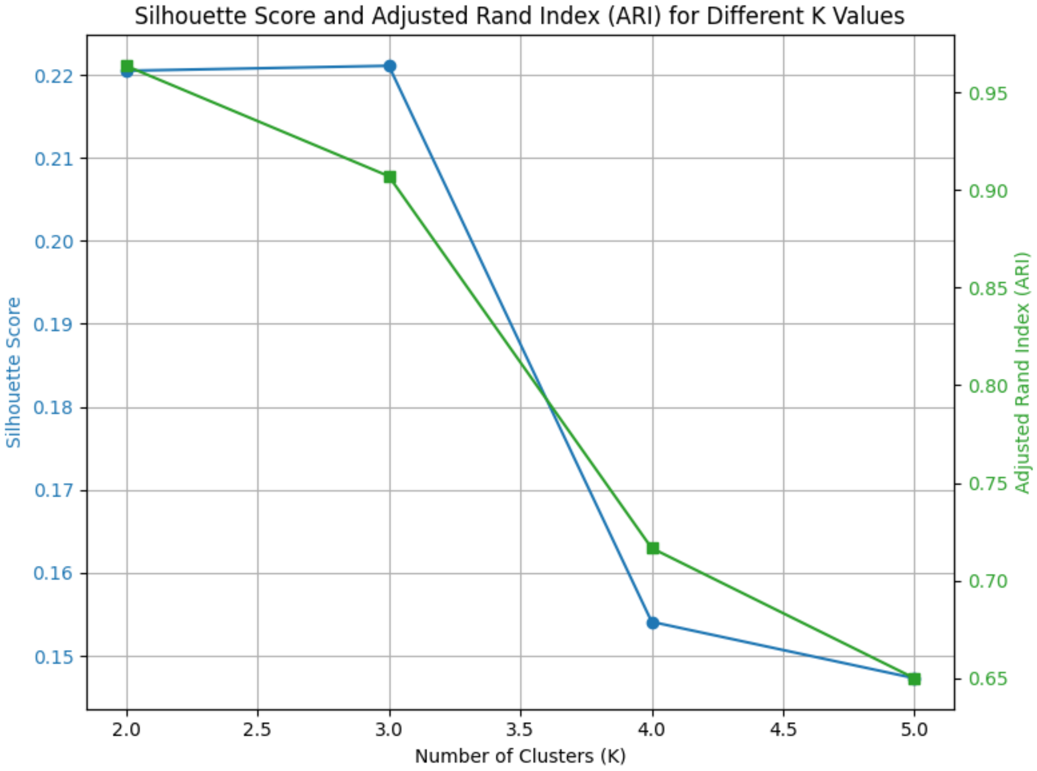


Fig. 23: Results of Silhouette Score & ARI For Clusters 2 - 5

Fig. 24 to 27 show visualisations of different clusters of k which support the scores which were found, k=2 produces the clearest clusters and as k increases there is less clarity. However, it can also be found that when k increases only the left cluster splits into different parts and the right one stays the same. This indicates that the legitimate URLs are more similar which is why they do not require further splitting. It provides a valuable insight that phishing URLs differ in the tactics used and can be evaluated to find which clusters could be potentially more sophisticated and dangerous. Also as seen in k=4, the yellow and blue cluster are closest to the border which can be useful to reach the objective.

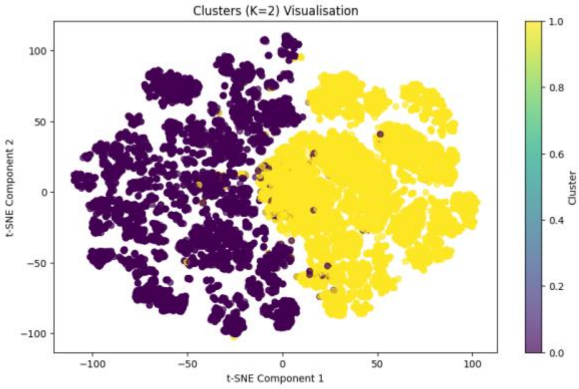


Fig. 24: k=2 t-SNE

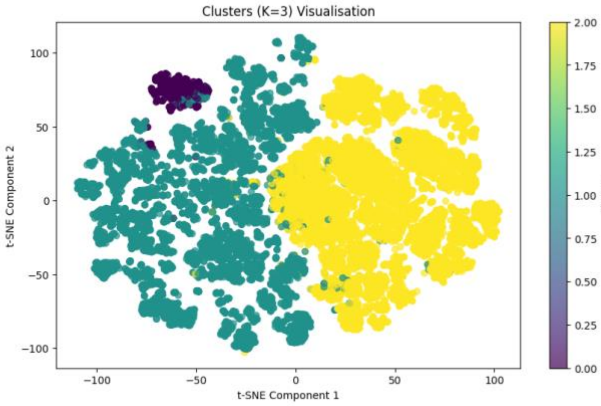


Fig. 25: k=3 t-SNE

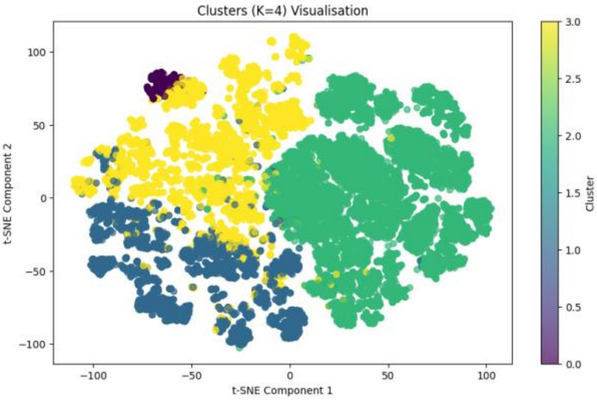


Fig. 26: k=4 t-SNE

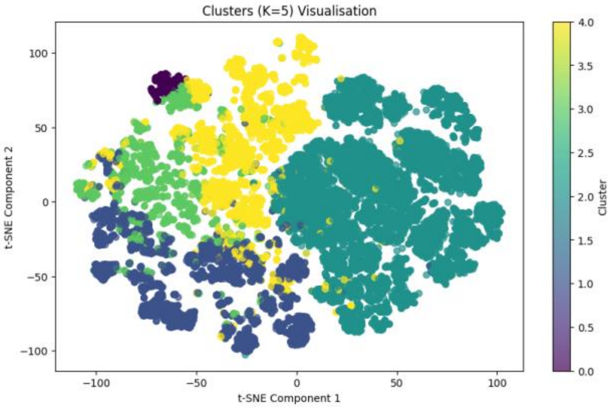


Fig. 27: k=5 t-SNE

ADDITIONAL FINDINGS

The SOM map (Fig. 28) indicates a few distinct clusters, the bright regions show separation whilst the dark show clusters of similar data. The bright areas can help analyse the boundaries of different types of data

The DEC (Fig. 29) shows a clear structure to the data where points next to each other belong to the same cluster, also showing some outliers as some points are far apart.

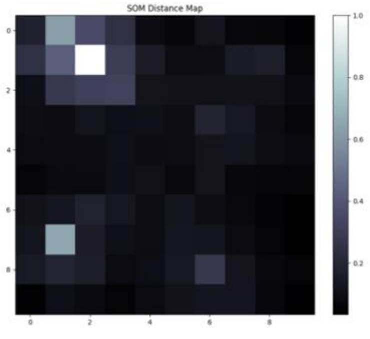


Fig. 28: SOM

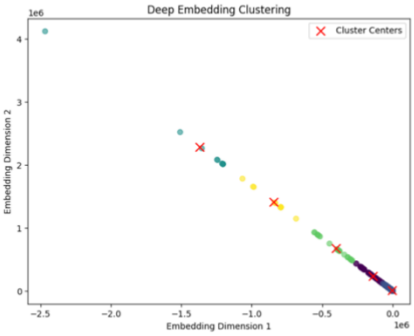


Fig. 29: DEC

Problem 3 (Unsupervised): Anomaly Detection

Research

Ripan et al., 2022:

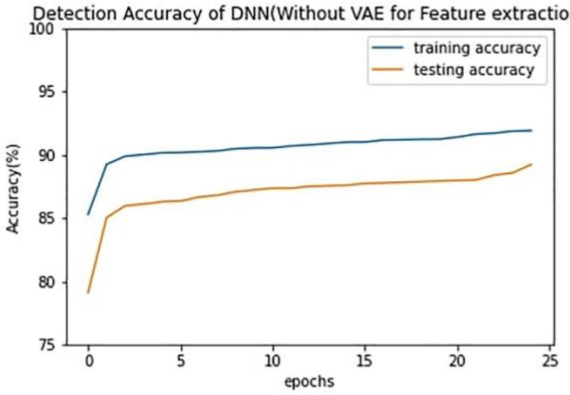
This research shows that isolation forest successfully identified outliers and performed better than DBSCAN and k-means. This led to improving the cyber-attack prediction model accuracy because of outlier removal.

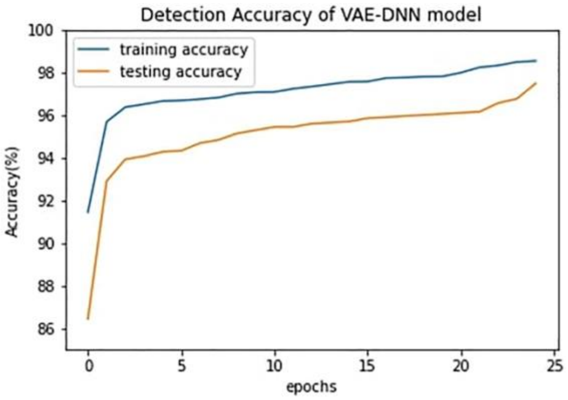
Somesha & Pais, 2022:

This study looks at combining word embedding techniques, such as Word2Vec and Fast Text, with machine learning algorithms to detect phishing emails. It presents accuracies scoring over 98% on various datasets and shows how implementing correct word embedding techniques can be beneficial.

Prabakaran et al., 2023:

This research paper uses the Variation Autoencoder to reduce the dimension of the input and extract important features. Then it uses a deep neural network to classify whether a URL is phishing. They used 25 training epochs; using VAE showed a significant improvement in detection accuracy.

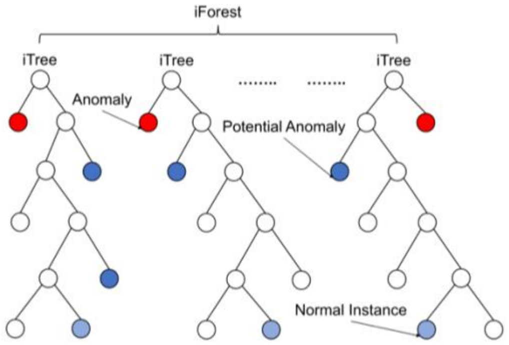




Machine learning algorithms

Isolation Forest (IF)

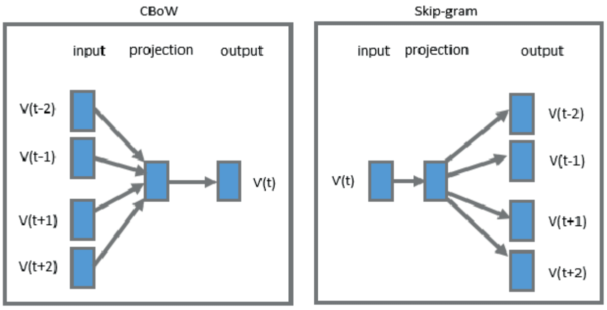
This is an unsupervised decision-tree based algorithm which was made to find anomalies. It splits data into random groups until there is a single point left. Outliers are usually easier to isolate (Cortes, n.d.).



(Regaya et al., 2021)

Word2Vec

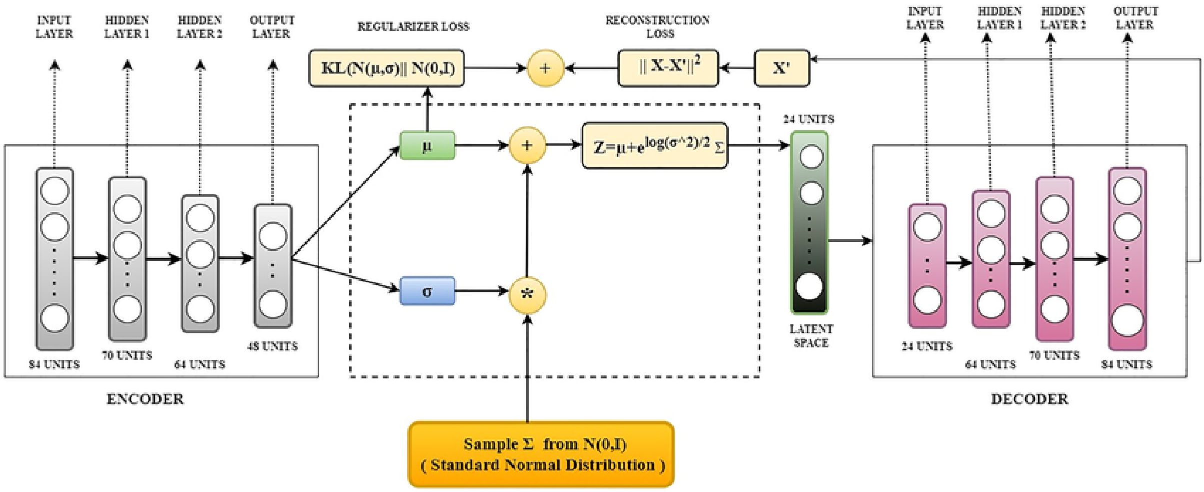
This is a model which looks at the relationships between words. It evaluates the words surrounding the target word and it generates meaningful embeddings for a given text (Somesha & Pais, 2022). CBoW is used in this report.



(Bilgin & Senturk, 2017)

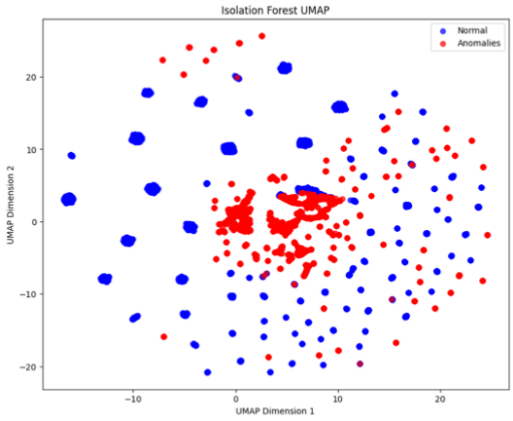
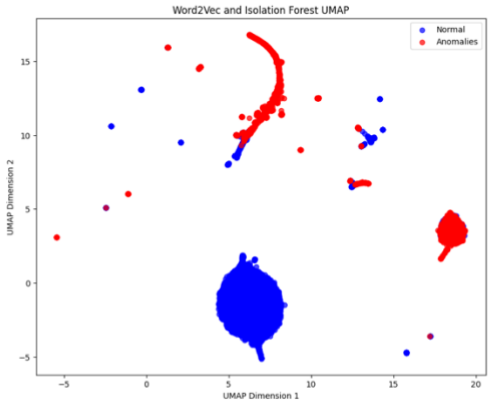
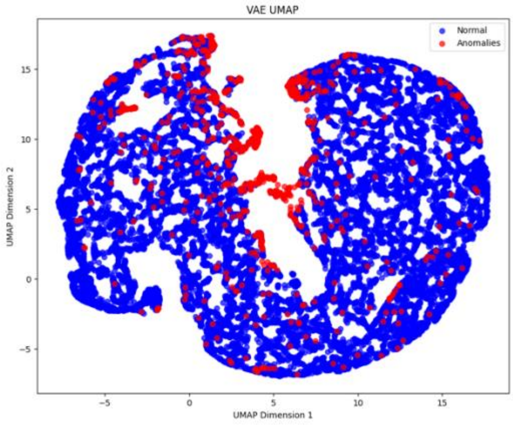
Variational Autoencoder (VAE)

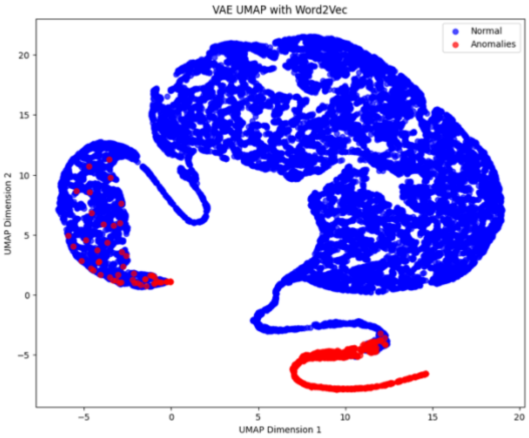
This is a neural network model which works by encoding and decoding data with the goal of creating a meaningful and compressed representation of the input and rebuilding the data to be as similar as possible to the input (Prabakaran et al., 2023).



(Prabakaran et al., 2023)

Results

Model	Silhouette Score	MSE	UMAP
IF	0.5372		
IF + Word2Vec	0.5261		
VAE	0.3112	1.0585	

VAE + Word2Vec	0.6728	0.9452	

It can be observed from the metrics that VAE with Word2Vec performed the best. Word2Vec is beneficial as it improved the silhouette score and lowered the mean squared error for VAE. The UMAP emphasises the better performance visually as the anomalies are further away from the normal data in contrast to the VAE UMAP and can be easily identified for further analysis. Word2Vec did not improve the silhouette score for IF however, the UMAP shows a better separation of anomalies. In the future other embedding techniques such as Fast Text should be used and hyperparameter optimisation to improve results.

CONCLUSION

This report looks at implementing machine learning algorithms to solve three problems within cybersecurity. These are: classifying phishing URLs, clustering URLs, and detecting anomalies. The first problem revealed that ensemble methods perform exceptionally, with XGB being on top. The clustering problem found that K-Means ++ offered the most balanced performance producing decent clusters that were true to real labels. Lastly, research for anomaly detection presented Word2Vec improved model performance. Consequently, this problem produced the lowest scores and needs more work. In the future using various algorithms together could produce better results.

DESCRIPTION	LINK
EDA	https://colab.research.google.com/drive/1TnLBz3HLha5M10R9OsDXr1WBexRooU6Y?usp=sharing
MACHINE LEARNING	https://colab.research.google.com/drive/1KpmrNqZm5nbluonCH2I6w4cfvgAZg4HE?usp=sharing

REFERENCES

Ahmed Soomro, A. et al. (2024) ‘Analysis of machine learning models and data sources to forecast burst pressure of petroleum corroded pipelines: A comprehensive review’, Engineering Failure Analysis, 155, p. 107747. doi:10.1016/j.engfailanal.2023.107747.

Ansari, M.F., Sharma, P.K. and Dash, B. (2022) ‘Prevention of phishing attacks using AI-based Cybersecurity Awareness Training’, International Journal of Smart Sensor and Adhoc Network., pp. 61–72. doi:10.47893/ijssan.2022.1221.

Bilgin, M. and Senturk, I.F. (2017) ‘Sentiment analysis on Twitter data with semi-supervised doc2vec’, 2017 International Conference on Computer Science and Engineering (UBMK), pp. 661–666. doi:10.1109/ubmk.2017.8093492.

Bishayee, S. (2023) A beginner’s Guide to Self Organizing Map (SOM) - deep learning, Medium. Available at: https://medium.com/@soumallya160/a-beginners-guide-to-self-organizing-map-som-deep-learning-e0e30a7534e3 (Accessed: 04 January 2025).

Chauhan, A. (2024) Random Forest classifier and its hyperparameters, Medium. Available at: https://medium.com/analytics-vidhya/random-forest-classifier-and-its-hyperparameters-8467bec755f6 (Accessed: 03 January 2025).

Chauhan, N. (2022) DBSCAN clustering algorithm in machine learning, KDnuggets. Available at: https://www.kdnuggets.com/2020/04/dbscan-clustering-algorithm-machine-learning.html (Accessed: 04 January 2025).

Chy, K. and Buadi, O. (2024) ‘A Machine Learning-Driven Website Platform and Browser Extension for Real-Time Risk Scoring and Fraud Detection for Website Legitimacy Verification and Consumer Protection’, Journal of Multidisciplinary Engineering Science and Technology (JMEST), 11(10).

Cortes, D. (no date) An introduction to isolation forests. Available at: https://cran.r-project.org/web/packages/isotree/vignettes/An_Introduction_to_Isolation_Forests.html (Accessed: 06 January 2025).

Cox, V. (2017) Translating statistics to make decisions: A guide for the non-statistician. New York: Apress.

Fawagreh, K., Gaber, M. M. and Elyan, E. (2014) ‘Random forests: from early developments to recent advancements’, Systems Science &amp; Control Engineering, 2(1), pp. 602–609. doi: 10.1080/21642583.2014.956265.

Gahlawat, N. (2024) Decision trees: A powerful tool in machine learning, Medium. Available at: https://medium.com/@nidhigahlawat/decision-trees-a-powerful-tool-in-machine-learning-dd0724dad4b6 (Accessed: 03 January 2025).

GeeksforGeeks (2023) XGBoost, GeeksforGeeks. Available at: https://www.geeksforgeeks.org/xgboost/ (Accessed: 03 January 2025).

Gov (no date) How to spot a fake website, Stop! Think Fraud. Available at: https://stopthinkfraud.campaign.gov.uk/how-to-spot-fraud/how-to-spot-a-fake-website/ (Accessed: 11 December 2024).

Haibo He and Garcia, E.A. (2009) 'Learning from Imbalanced Data', IEEE Transactions on Knowledge and Data Engineering, 21(9), pp. 1263–1284. doi:10.1109/tkde.2008.239.

Inyang, U.G. (2021) Unsupervised characterization and visualization of students' academic performance features, Computer and Information Science. Available at: https://www.academia.edu/51325574/Unsupervised_Characterization_and_Visualization_of_Students_Academic_Performance_Features (Accessed: 04 January 2025).

Kara, I., Ok, M. and Ozaday, A. (2022) 'Characteristics of understanding urls and domain names features: The detection of phishing websites with Machine Learning Methods', IEEE Access, 10, pp. 124420–124428. doi:10.1109/access.2022.3223111.

Ke, G. et al. (no date) LightGBM: A highly efficient gradient boosting decision tree, Advances in Neural Information Processing Systems. Available at: https://papers.nips.cc/paper/6907-lightgbm-a-highly-efficient-gradient-boosting-decision (Accessed: 03 January 2025).

Kok, J.W.T.M. et al. (2024) Deep embedded clustering generalisability and adaptation for integrating mixed datatypes: Two critical care cohorts, Nature News. Available at: https://www.nature.com/articles/s41598-024-51699-z (Accessed: 04 January 2025).

Komorowski, M. et al. (2016) 'Exploratory Data Analysis', in Secondary Analysis of Electronic Health Records. Cham: Springer, pp. 185–203.

Korkmaz, M., Sahingoz, O.K. and Diri, B. (2020) 'Detection of phishing websites by using Machine Learning-based URL analysis', 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT) [Preprint]. doi:10.1109/icccnt49239.2020.9225561.

Korkmaz, M., Sahingoz, O.K. and Diri, B. (2020) 'Detection of phishing websites by using Machine Learning-based URL analysis', 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT) [Preprint]. doi:10.1109/icccnt49239.2020.9225561.

Kumar, N. (2023) 'Detection of textual propaganda using passive aggressive classifiers', International Journal of Advanced Trends in Computer Science and Engineering, 12(2), pp. 73–79. doi:10.30534/ijatcse/2023/071222023.

M., S. and Pais, A.R. (2022) 'Classification of phishing email using word embedding and machine learning techniques', Journal of Cyber Security and Mobility [Preprint]. doi:10.13052/jcsm2245-1439.1131.

Mandot, P. (2017) What is LIGHTGBM, how to implement it? how to fine tune the parameters?, Medium. Available at: https://medium.com/@pushkarmandot/https-medium-com-pushkarmandot-what-is-lightgbm-how-to-implement-it-how-to-fine-tune-the-parameters-60347819b7fc (Accessed: 03 January 2025).

Mohammad, R., Thabtah, F. and McCluskey, L. (2015) Phishing Websites Features. Available at: https://eprints.hud.ac.uk/id/eprint/24330/6/MohammadPhishing14July2015.pdf (Accessed: 13 December 2024).

Moore, A. and Bell, M. (2022) XGBoost, a novel explainable AI technique, in the prediction of myocardial infarction, a UK Biobank Cohort Study [Preprint]. doi:10.1101/2022.04.08.22273600.

Ng, C.K. et al. (2019) 'Static malware clustering using enhanced deep embedding method', Concurrency and Computation: Practice and Experience, 31(19). doi:10.1002/cpe.5234.

Office for National Statistics (2022) Phishing attacks – who is most at risk? Available at: https://www.ons.gov.uk/peoplepopulationandcommunity/crimeandjustice/articles/phishingattackswhoismostatrisk/2022-09-26 (Accessed: 11 December 2024).

Ojewumi, T. et al. (2022) Performance evaluation of machine learning tools for detection of phishing attacks on web pages, Scientific African. Available at: https://www.sciencedirect.com/science/article/pii/S2468227622000746 (Accessed: 14 December 2024).

Oluleye, A. (2023) Exploratory Data Analysis with python cookbook: Over 50 recipes to analyze, visualize, and extract insights from structured and unstructured data. Birmingham: Packt Publishing.

Ontivero-Ortega, M. et al. (2017) 'Fast gaussian naïve bayes for Searchlight Classification Analysis', NeuroImage, 163, pp. 471–479. doi:10.1016/j.neuroimage.2017.09.001.

Petrosyan, A. (2024) Global Cybercrime estimated cost 2029, Statista. Available at: https://www.statista.com/forecasts/1280009/cost-cybercrime-worldwide (Accessed: 11 December 2024).

Prabakaran, M.K., Meenakshi Sundaram, P. and Chandrasekar, A.D. (2023) 'An enhanced deep learning-based phishing detection mechanism to effectively identify malicious urls using variational autoencoders', IET Information Security, 17(3), pp. 423–440. doi:10.1049/ise2.12106.

Prabakaran, M.K., Meenakshi Sundaram, P. and Chandrasekar, A.D. (2023) 'An enhanced deep learning-based phishing detection mechanism to effectively identify malicious urls using variational autoencoders', IET Information Security, 17(3), pp. 423–440. doi:10.1049/ise2.12106.

Prasad, A. and Chandra, S. (2024) Phiusiil: A diverse security profile empowered phishing URL detection framework based on similarity index and Incremental Learning, Computers & Security. Available at: https://www.sciencedirect.com/science/article/pii/S0167404823004558?via%3Dihub (Accessed: 11 December 2024).

Prateek (2022) K means clustering algorithm, KeyToDataScience. Available at: https://keytodatascience.com/k-means-clustering-algorithm/ (Accessed: 04 January 2025).

Radi, K. and Moughit, M. (2023) Enhancing cybersecurity through clustering analysis of DNS queries, HAL Open science. Available at: https://hal.science/hal-04219804v1/document (Accessed: 04 January 2025).

Radi, K. and Moughit, M. (2023) Enhancing cybersecurity through clustering analysis of DNS queries, HAL Open science. Available at: https://hal.science/hal-04219804v1/document (Accessed: 04 January 2025).

Re, M. (2023) Hierarchical clustering: A comprehensive guide to understanding and applying this powerful data analysis technique, LinkedIn. Available at:

https://www.linkedin.com/pulse/hierarchical-clustering-comprehensive-guide-understanding-massimo-re-r73gf (Accessed: 04 January 2025).

Regaya, Y., Fadli, F. and Amira, A. (2021) 'Point-denoise: Unsupervised outlier detection for 3D point clouds enhancement', Multimedia Tools and Applications, 80(18), pp. 28161–28177. doi:10.1007/s11042-021-10924-x.

Ripan, R.C. et al. (2022) 'Effectively predicting cyber-attacks through Isolation Forest learning-based outlier detection', SECURITY AND PRIVACY, 5(3). doi:10.1002/spy2.212.

Sandhya, V.P., Vanila, S., Durgadevi, K. and Sobana, K., Predictive Methods For Heart Disease Using Bivariate Analysis. Computer Integrated Manufacturing Systems, 1006, p.5911.

Scikit (no date) RFE, Scikit Learn. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFE.html (Accessed: 07 January 2025).

Shyrokykh, K., Girnyk, M. and Dellmuth, L. (2023) 'Short text classification with machine learning in the Social Sciences: The case of climate change on Twitter', PLOS ONE, 18(9). doi:10.1371/journal.pone.0290762.

Torres, L. (2023) XGBoost: The King of Machine Learning Algorithms, Medium. Available at: https://medium.com/latinxinai/xgboost-the-king-of-machine-learning-algorithms-6b5c0d4acd87#:~:text=XGBoost%20was%20first%20introduced%20by,with%20higher%20accuracy%20for%20predictions. (Accessed: 03 January 2025).

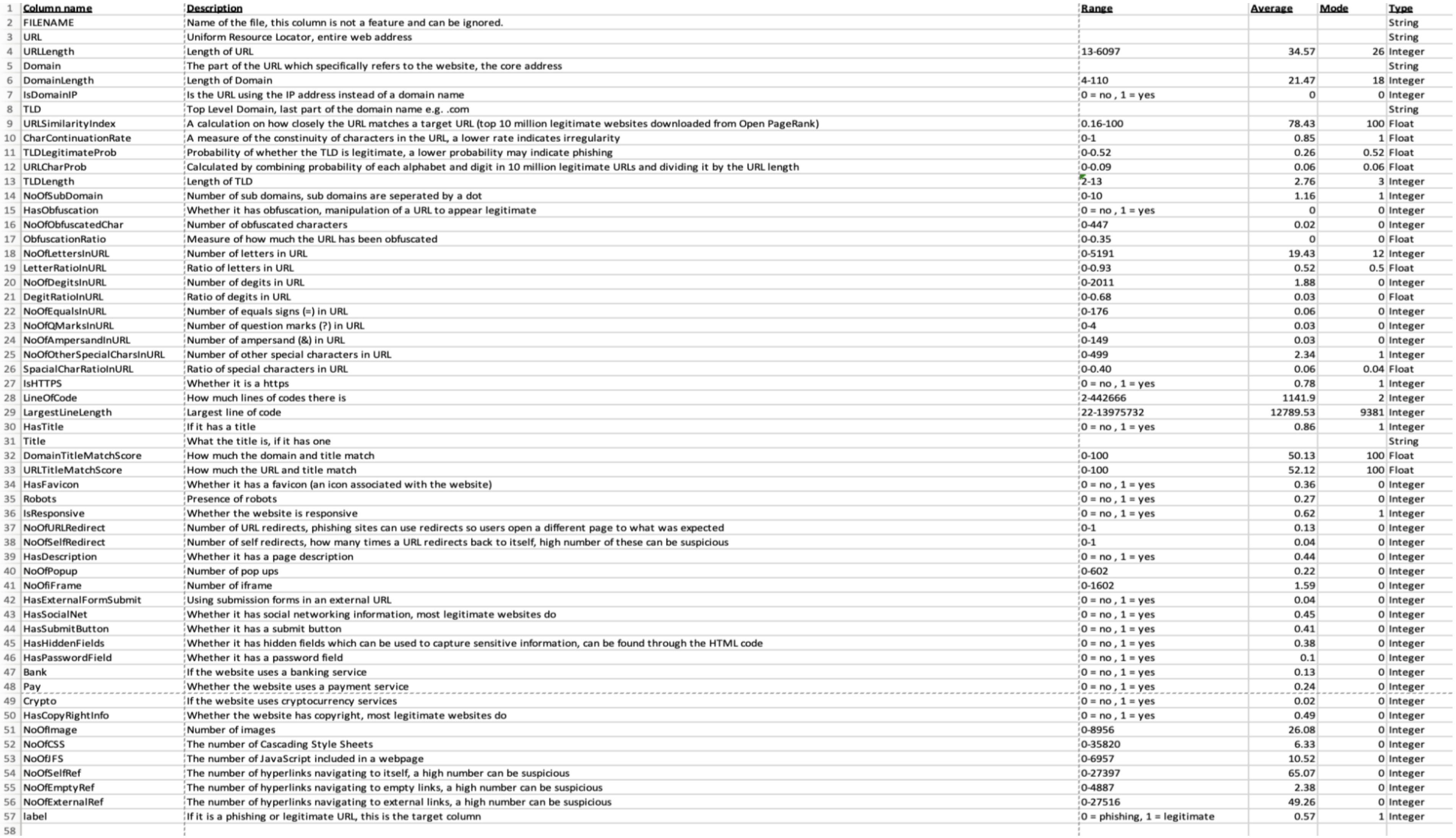
Ullrich, J. (2023) The .zip gtld: Risks and opportunities - sans internet storm center, SANS Technology Institute. Available at: https://isc.sans.edu/diary/The+zip+gTLD+Risks+and+Opportunities/29838 (Accessed: 12 December 2024).

Ville, B. de (2013) 'Decision trees', WIREs Computational Statistics, 5(6), pp. 448–455. doi:10.1002/wics.1278.

Xie, J., Girshick, R. and Farhadi, A. (2016) 'Unsupervised Deep Embedding for Clustering Analysis', Proceedings of The 33rd International Conference on Machine Learning [Preprint].

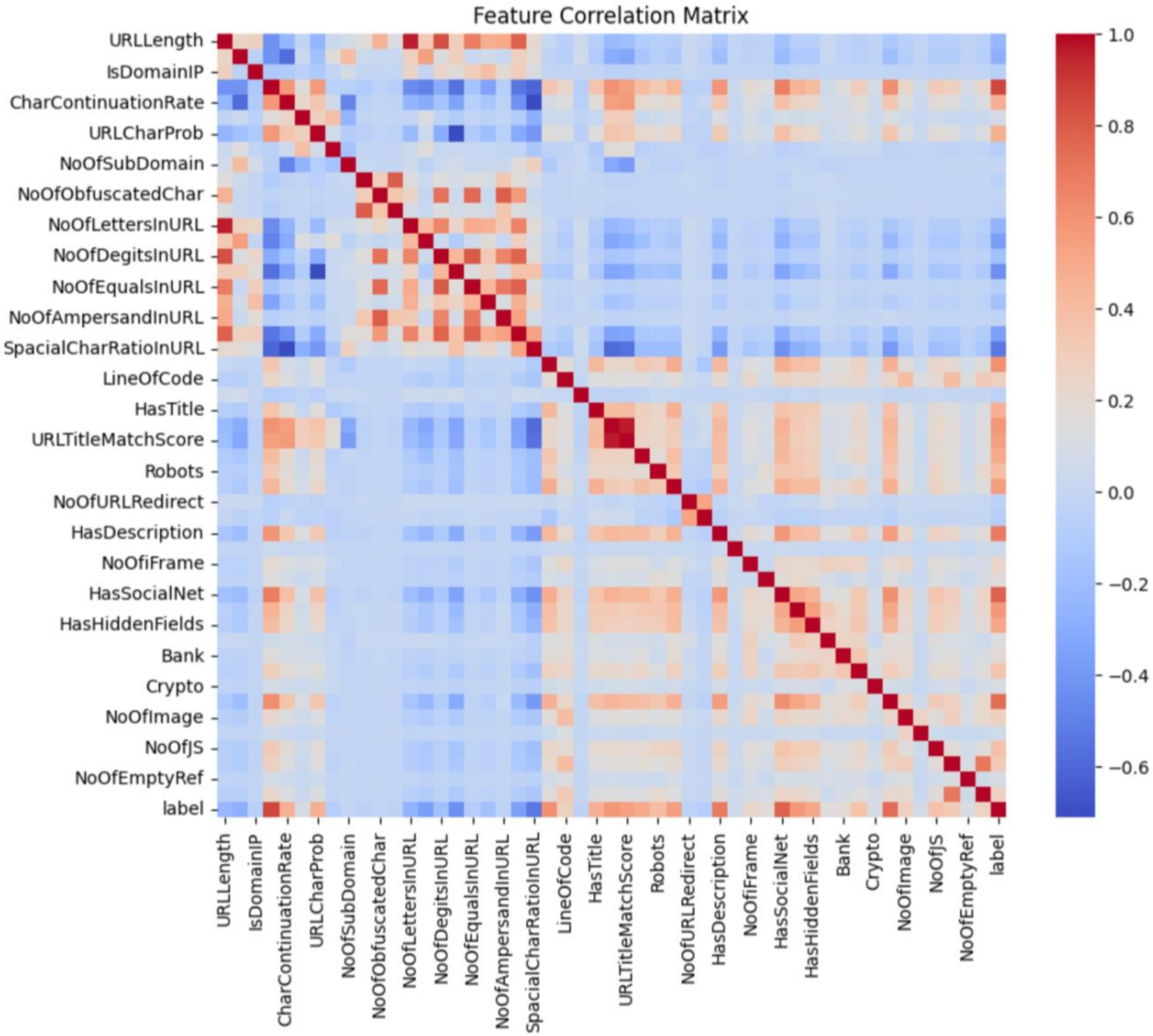
APPENDICES

APPENDIX 1

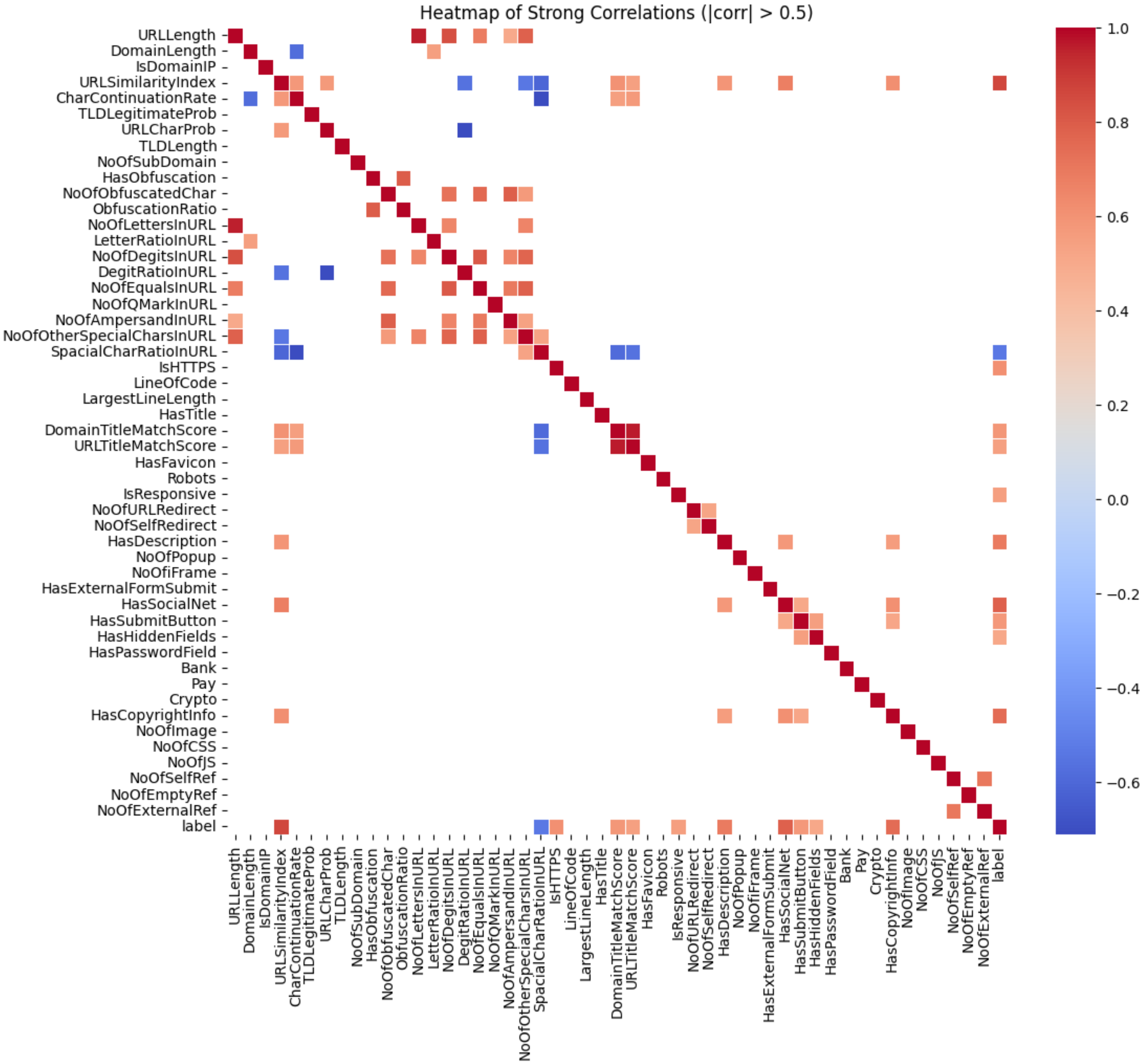


Column name	Description	Range	Average	Mode	Type
FILENAME	Name of the file, this column is not a feature and can be ignored.				String
URL	Uniform Resource Locator, entire web address				String
URLLength	Length of URL	13-6097	34.57	26	Integer
Domain	The part of the URL which specifically refers to the website, the core address				String
DomainLength	Length of Domain	4-110	21.47	18	Integer
IsDomainIP	Is the URL using the IP address instead of a domain name	0 = no , 1 = yes	0	0	Integer
TLD	Top Level Domain, last part of the domain name e.g. .com				String
URLSimilarityIndex	A calculation on how closely the URL matches a target URL (top 10 million legitimate websites downloaded from Open PageRank)	0.16-100	78.43	100	Float
CharContinuationRate	A measure of the constinuity of characters in the URL, a lower rate indicates irregularity	0-1	0.85	1	Float
TLDLegitimateProb	Probability of whether the TLD is legitimate, a lower probability may indicate phishing	0-0.52	0.26	0.52	Float
URLCharProb	Calculated by combining probability of each alphabet and digit in 10 million legitimate URLs and dividing it by the URL length	0-0.09	0.06	0.06	Float
TLDLength	Length of TLD	2-13	2.76	3	Integer
NoOfSubDomain	Number of sub domains, sub domains are seperated by a dot	0-10	1.16	1	Integer
HasObfuscation	Whether it has obfuscation, manipulation of a URL to appear legitimate	0 = no , 1 = yes	0	0	Integer
NoOfObfuscatedChar	Number of obfuscated characters	0-447	0.02	0	Integer
ObfuscationRatio	Measure of how much the URL has been obfuscated	0-0.35	0	0	Float
NoOfLettersInURL	Number of letters in URL	0-5191	19.43	12	Integer
LetterRatioInURL	Ratio of letters in URL	0-0.93	0.52	0.5	Float
NoOfDegitsInURL	Number of degits in URL	0-2011	1.88	0	Integer
DegitRatioInURL	Ratio of degits in URL	0-0.68	0.03	0	Float
NoOfEqualsInURL	Number of equals signs (=) in URL	0-176	0.06	0	Integer
NoOfQMarksInURL	Number of question marks (?) in URL	0-4	0.03	0	Integer
NoOfAmpersandInURL	Number of ampersand (&) in URL	0-149	0.03	0	Integer
NoOfOtherSpecialCharsInURL	Number of other special characters in URL	0-499	2.34	1	Integer
SpacialCharRatioInURL	Ratio of special characters in URL	0-0.40	0.06	0.04	Float
IsHTTPS	Whether it is a https	0 = no , 1 = yes	0.78	1	Integer
LineOfCode	How much lines of codes there is	2-442666	1141.9	2	Integer
LargestLineLength	Largest line of code	22-13975732	12789.53	9381	Integer
HasTitle	If it has a title	0 = no , 1 = yes	0.86	1	Integer
Title	What the title is, if it has one				String
DomainTitleMatchScore	How much the domain and title match	0-100	50.13	100	Float
URLTitleMatchScore	How much the URL and title match	0-100	52.12	100	Float
HasFavicon	Whether it has a favicon (an icon associated with the website)	0 = no , 1 = yes	0.36	0	Integer
Robots	Presence of robots	0 = no , 1 = yes	0.27	0	Integer
IsResponsive	Whether the website is responsive	0 = no , 1 = yes	0.62	1	Integer
NoOfURLRedirect	Number of URL redirects, phishing sites can use redirects so users open a different page to what was expected	0-1	0.13	0	Integer
NoOfSelfRedirect	Number of self redirects, how many times a URL redirects back to itself, high number of these can be suspicious	0-1	0.04	0	Integer
HasDescription	Whether it has a page description	0 = no , 1 = yes	0.44	0	Integer
NoOfPopup	Number of pop ups	0-602	0.22	0	Integer
NoOfiFrame	Number of iframe	0-1602	1.59	0	Integer
HasExternalFormSubmit	Using submission forms in an external URL	0 = no , 1 = yes	0.04	0	Integer
HasSocialNet	Whether it has social networking information, most legitimate websites do	0 = no , 1 = yes	0.45	0	Integer
HasSubmitButton	Whether it has a submit button	0 = no , 1 = yes	0.41	0	Integer
HasHiddenFields	Whether it has hidden fields which can be used to capture sensitive information, can be found through the HTML code	0 = no , 1 = yes	0.38	0	Integer
HasPasswordField	Whether it has a password field	0 = no , 1 = yes	0.1	0	Integer
Bank	If the website uses a banking service	0 = no , 1 = yes	0.13	0	Integer
Pay	Whether the website uses a payment service	0 = no , 1 = yes	0.24	0	Integer
Crypto	If the website uses cryptocurrency services	0 = no , 1 = yes	0.02	0	Integer
HasCopyRightInfo	Whether the website has copyright, most legitimate websites do	0 = no , 1 = yes	0.49	0	Integer
NoOfImage	Number of images	0-8956	26.08	0	Integer
NoOfCSS	The number of Cascading Style Sheets	0-35820	6.33	0	Integer
NoOfJS	The number of JavaScript included in a webpage	0-6957	10.52	0	Integer
NoOfSelfRef	The number of hyperlinks navigating to itself, a high number can be suspicious	0-27397	65.07	0	Integer
NoOfEmptyRef	The number of hyperlinks navigating to empty links, a high number can be suspicious	0-4887	2.38	0	Integer
NoOfExternalRef	The number of hyperlinks navigating to external links, a high number can be suspicious	0-27516	49.26	0	Integer
label	If it is a phishing or legitimate URL, this is the target column	0 = phishing, 1 = legitimate	0.57	1	Integer

APPENDIX 2


Feature Correlation Matrix
URLLength
IsDomainIP
CharContinuationRate
URLCharProb
NoOfSubDomain
NoOfObfuscatedChar
NoOfLettersInURL
NoOfDegitsInURL
NoOfEqualsInURL
NoOfAmpersandInURL
SpacialCharRatioInURL
LineOfCode
HasTitle
URLTitleMatchScore
Robots
NoOfURLRedirect
HasDescription
NoOfiFrame
HasSocialNet
HasHiddenFields
Bank
Crypto
NoOfImage
NoOfJS
NoOfEmptyRef
label
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6

APPENDIX 3


Heatmap of Strong Correlations (|corr| > 0.5)
URLLength
DomainLength
IsDomainIP
URLSimilarityIndex
CharContinuationRate
TLDLegitimateProb
URLCharProb
TLDLength
NoOfSubDomain
HasObfuscation
NoOfObfuscatedChar
ObfuscationRatio
NoOfLettersInURL
LetterRatioInURL
NoOfDegitsInURL
DegitRatioInURL
NoOfEqualsInURL
NoOfQMarkInURL
NoOfAmpersandInURL
NoOfOtherSpecialCharsInURL
SpacialCharRatioInURL
IsHTTPS
LineOfCode
LargestLineLength
HasTitle
DomainTitleMatchScore
URLTitleMatchScore
HasFavicon
Robots
IsResponsive
NoOfURLRedirect
NoOfSelfRedirect
HasDescription
NoOfPopup
NoOfiFrame
HasExternalFormSubmit
HasSocialNet
HasSubmitButton
HasHiddenFields
HasPasswordField
Bank
Pay
Crypto
HasCopyrightInfo
NoOfImage
NoOfCSS
NoOfJS
NoOfSelfRef
NoOfEmptyRef
NoOfExternalRef
label
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6