National College of Ireland

Detection of Phishing Websites Using Machine Learning

MSc Research Project
MSc Cybersecurity

Akash Muralidharan
Student ID: 23233192

School of Computing
National College of Ireland

Supervisor: Vikas Sahni

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Akash Muralidharan		
Student ID:	23233192		
Programme:	MSc Cyber Security	**Year:**	2024-2025
Module:	MSc Practicum/Internship part 2		
Supervisor:	Vikas Sahni		
Submission Due Date:	29th January 2025		
Project Title:	Detection of Phishing Websites Using Machine Learning		
Word Count:	5796		

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

Signature: Akash Muralidharan

Date: 29th January 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

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

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

Detection of Phishing Websites Using Machine Learning

Akash Muralidharan
23233192

Abstract

Phishing attacks pose a significant threat to both corporate and personal internet users, compromising personal data and financial security. As attackers evolve their tactics, there is an urgent need for updated detection strategies. Machine learning models offer distinct advantages in detecting phishing attacks by learning patterns from large datasets and therefore providing more accurate and efficient predictions of evolving threats than traditional rule-based methods. This research examines how machine learning and AI can help prevent phishing attacks and by analysing website features, these models can effectively identify malicious sites, offering a more accurate and adaptive approach to detecting phishing threats. The performance of these models was evaluated on metrics like accuracy, precision, recall and F1 score, and this was done to obtain a better analysis of how these models perform. The results suggest that ensemble methods such as Random Forest are robust which gives high precision and recall for the prediction of phishing efforts. The system demonstrates the possibility to adapt to a variety of phishing patterns, and scales efficiently thanks to preprocessing. This solution represents an attainable and scalable phishing detection framework that promises practical action for organizations toward mitigating the defense against ever changing threats. Though the results are promising, more work is needed to integrate real time detection capabilities and extend the evaluation across larger and dynamic datasets. A huge gain in detection accuracy is also achieved by exploring deep learning models in this case.

Keywords – Phishing, Machine Learning, Logistic Regression, Random Forest Classifier.

1 Introduction

Phishing attacks are perhaps the most common and harmful form of cybercrime and present a serious risk to individuals, organizations and governments around the world. In an age when

the web is increasing and constantly changing, so are the methods malicious hackers employ to trick users into handing over their private information, like login information and credit card details. Traditional ways of phishing detection such as rule based systems and blacklists do not suffice when it comes to curbing sophisticated and dynamic current era phishing attack. As such, researchers are being relied upon Machine Learning (ML), and Artificial Intelligence (AI) models to tackle this urgent problem. This new arsenal of advanced techniques promises to find the intricate patterns within data and improve responding to any new phishing threat.

Phishing attacks do not just cost money, but they also erode user trust, harm the reputation of organizations. Existing phishing detection systems struggle to keep pace with new phishing techniques, threatening to undermine the security of the user and business alike. Machine Learning and AI models promise to be an alternative that is able to detect threats more efficiently and adapt to new emerging threats. But accurately and reliably producing these models, while making them adaptable, is very important and needs to be investigated in detail. ML & AI provides a huge opportunity to improve the phishing detection systems but doing that requires a solid approach to the research as well as implementation.

Several variables, however, act as the determinants of the success of ML and AI models in phishing prevention. They amount to the quality and diversity of the set of datasets used in training, the algorithms being used and the features engineered for detection. Additionally, since these models need to adapt to constantly shifting threat environments and must correctly differentiate between phishing attempts and genuine communication, these factors contribute significantly to the success of these models. To build a system that is both reliable and scalable, it is essential to address these variables.

The research question addressed in this study is: "How Machine Learning and AI models can help prevent phishing attacks, and how can it be done accurately?" The objective in this study includes phishing detection using ML/AI techniques and to implement the same suing real-world datasets for producing a highly accurate model. Then, the final objective is to evaluate the performance of the model in terms of accuracy, adaptability and efficiency. However, this research has limitations. Primarily, it is centered on AI based detection system, and it might not cover all phishing aspects, For example, social engineering. Moreover, its effectiveness is based on the quality and variety of the training data that could affect its generalizability.

This work contributes to the state of the art by developing an ML/AI based framework that improves phishing detection accuracy and dynamically adapts to phishing threats, as they evolve. This study helps bridge the phasing gap between traditional methods and advanced AI driven solutions therefore resulting in a practical and impact full approach for dealing with the global challenge of phishing attacks.

This report starts with a comprehensive literature review of existing methods to detect phishing and artificial intelligence models. Next, the proposed system is designed and implemented through a methodology section. Afterwards, the results section describes evaluating the system performance and the next section discusses implications, challenges and future work. Finally,

the conclusion presents some final key findings and their implications in the wider cybersecurity domain.

2 Related Works

2.1 Similar Works

Rishikesh Mahajan (2018) dealt with different machine learning algorithms like K-Nearest Neighbours (KNN), Naive Bayes, Decision Tree, and Gradient Boosting in detecting phishing websites. Some URL based features like the presence of symbols in URL and length of the URL can be used for identifying phishing sites. Also, domain-based features like page rank and domain age help in the same. On analyzing different algorithms, the Decision Tree was shown the best performance with an F1 score of 0.94 which strongly indicates high accuracy and hence better classification, preventing such phishing attacks from happening.

Aniket Garje (2021) applies algorithms in machine learning such as KNN, Naive Bayes, Gradient Boosting and Decision Tree to detect phishing websites. This focuses on feature extraction from the URL and domains characteristics such as symbols like @, subdomains and domain age. Data was split into training and testing sets (80:20). The performance of these algorithms was evaluated on the basis of precision, recall, and F1 score. Amongst all the algorithms, the Decision Tree algorithm showed good performance overall giving just the right balance of precision and recall and hence was chosen to be a good choice to detect phishing websites.

Ashit Kumar Dutta (2021) performed a study to detect phishing websites by using Recurrent Neural Networks (RNN) with Long ShortTerm Memory (LSTM). It utilizes Phishtank and AlexaRank datasets, classifying malicious websites versus legitimate ones, by utilizing the features extracted from URLs and site content. The results further show that the proposed LSTM based approach has a superior accuracy, precision, and F1 scores compared with existing methods, capable of dealing with large amount of data. This work explores how advanced machine learning techniques can be useful against phishing attacks, and future work focuses on unsupervised techniques as well as scalability.

Ozgur Koray Sahingoz (2019) suggests a phishing URL detection system that is based on seven classifiers such as Random Forest, Naïve Bayes, Decision Tree and utilizes NLP based features. Using a dataset of 73,575 URLs, the system reported the highest accuracy of 97.98% using Random Forest classifier. It also focuses on good features like language independent, real time detection. The study shows that by combining feature rich classifiers with hybrid approaches, phishing websites can be detected better, and the future work will focus on combining deep learning techniques to boost accuracy and scalability.

Joby James (2013) shows methods to detect phishing websites are discussed based on lexical, host based and page importance features. It analyses the performance of four machine

learning algorithms namely Naïve Bayes, J48 Decision Tree, K-NN and SVM by applying them on a phishing and benign URLs dataset to evaluate the performance of these algorithms in WEKA and MATLAB environments. One achieving the highest success rate of 93.2% of detection accuracy was the J48 Decision Tree. The research highlights that to improve phishing detection, efficient feature extraction and classification algorithms should be used and future directions involve improving online learning techniques to accommodate everchanging phishing tactics.

Pradeepthi K V (2014) performed analysis of different classification algorithms for phishing URL detection using a dataset of 4,500 URLs (2,500 genuine and 2,000 phishing). The study analyzes lexical, URL based, network, and domain features and shows that tree based classifiers Random Forest and Random Tree achieve the highest accuracy, precision and recall as close as to 99%. The paper claims that tree based methods are effective for phishing detection and suggests future work in that will incorporate online learning mechanisms to improve adaptability and accuracy in the changing relational environment.

Aniket Garje (2021) describes about the detection of phishing websites using machine learning algorithms including Random Forest, SVM, Neural networks and Decision Trees. It achieves high detections accuracy (up to 98%) by exploiting lexical and domain specific features, URL structure, and page content, using advanced feature selection techniques. It was stressed that tree-based models (especially Random Forests) are a very robust and precise family of algorithms that have outperformed other algorithms in the study. This also demonstrates the integration of these methods into real world applications (browser extensions) for improving cybersecurity against phishing attacks.

R. Kiruthiga (2019) outlines web phishing detection using machine learning techniques. It applied each of a number of algorithms (Random Forest, Decision tree, Gradient boosting, SVM) on features (URL structure, domain attributes, heuristic rules) to identify phishing sites. The Random Forest Model performed well in the task, and it performed the best with high accuracy (up to 98.4%) and with robust precision and recall. The significance of feature selection and optimization to enhancing detection programs, as well as novel schemes such as PhishScore and PhishChecker are also emphasized. The growth in sophistication of phishing attacks emphasizes the need for adaptive methods of combating them.

2.2 Recent Works

Yahia Said (2024) proposed a phishing URL detection model through an improved version of Convolutional Neural Networks (CNN) combined with a multi-head self-attention mechanism. Generative Adversarial Networks (GAN) is employed to generate phishing URLs to solve the problem of imbalanced datasets, and make the model learn on balanced training data. Experimental results show that the proposed model made quantitative gain in accuracy, precision, recall, and F1 score over the latter. Finally, the study successfully combines self-attention mechanisms with CNNs, which leads to higher detection accuracy and reliable performance for real world datasets.

Najwa Altwaijry (2024) studied the effectiveness of using 1D Convolutional Neural Networks (1D-CNN) together with recurrent layers (LSTM, Bi-LSTM, GRU, Bi-GRU) for phishing email detection. The results of the study are found using benchmark datasets such as Phishing Corpus and Spam Assassin showing that best results are achieved by 1D-CNN augmented with Bi-GRU having 99.68% accuracy and high F1 score of 99.66%. The paper also emphasizes advantages of deep learning for automating feature extraction to overcome deficiencies of traditional machine learning techniques and for protecting against phishing attacks. However, it is focused on lightweight, high performance models and suggests future research into expanding datasets and applying advanced algorithms to future improved detection systems.

Sri Hari Nallamala (2024) proposed a research where the Gradient Boosting Classifier, Random Forest and Decision Tree algorithms were used to discover the phishing URLs. It evaluates the performance of these models on a dataset of 11,054 URLs using two feature sets: 30 attribute comprehensive set and a feature key set of 13 attributes was selected by using SelectKBest and ChiSquare methods. It was found that the best accuracy of 97.4% is achieved with the complete dataset, and for 95.6% accuracy the key feature dataset is better with a better interpretability of the model. The results also highlight the benefit of feature selection in enhancing efficiency and providing real time phishing detector solutions for web browsers and security applications.

G. K. Kamalam (2024) suggested in this paper that it is a comprehensive study on how deep learning and AI models help detect malicious URLs which help prevent phishing attacks. The discussion also comprises of different machine learning techniques and also most importantly deep reinforcement learning in which the model learns adaptively from the dynamic nature of phishing websites. This helps further improve the accuracy of the model. Hence this study not only focusses on the identification of phishing websites but also emphasizes the importance of continuous learning and adaptation to the evolving threats in the digital world.

Kateryna Burbela (2023) gives a comprehensive approach towards the detection of phishing URLs. It proposes an innovative model that combines Convolutional Neural Networks (CNN) and Multi-Head Self-Attention (MHSA) for obtaining better accuracy for the detection process. Also, the study shows that the CNN-MHSA ensemble model achieves an accuracy rate of 98.3%. This hybrid model approach not only improves the accuracy of detection but also showcases the adaptability of machine learning algorithms in the evolving phishing threats.

Mahdi Bahaghighat (2023) studied where the need for accurate and fast classification algorithms is necessary. The research utilized various classification algorithms including Logistic Regression, K-Nearest Neighbors, Naive Bayes, Random Forest, Support Vector Machine, and Extreme Gradient Boosting (XGBoost), on a dataset of 88,647 instances (58,000 legitimate and 30,647 phishing) for the identification of phishing websites. The XGBoost was

effectively and accurately classifying the websites with an accuracy of 99.22% and excellent precision, recall, F1 score, and specificity. Hence the study shows that Machine 4 learning algorithms with feature selection and data balancing techniques like SMOTEENN enhance the detection accuracy making it viable to prevent phishing attacks.

Shouq Alnemari (2023) discusses the use of Machine Learning models to improve detection of phishing websites .The performance of models such as Random Forest (RF), Support Vector Machine (SVM), Artificial Neural Networks (ANN), and Decision Trees (DT) is evaluated using the UCI phishing domain dataset. In order to increase the accuracy, the study shows that the technique of feature selection and MinMax normalization is used on the Random Forest model to achieve the highest accuracy (97.3%). The study emphasizes the increasing roles of machine learning in cybersecurity and compares algorithms for effective phishing threat confrontation. Future work will explore further algorithms to improve on detection systems.

2.3 Gaps in the Literature and Conclusions

However, the reviewed literature illustrates the important progress made in phishing detection, using ML and DL approaches. Yet, there are still many notable gaps. In spite of the fact that a significant number of studies have emphasized on high accuracy in controlled environments, they pay little or no attention to real world adaptability specifically with respect to handling dynamic and evolving phishing attacks. Specific datasets are dependent, and the models are not often generalizable across very different domains. There are few studies that investigate unsupervised learning or reinforcement learning for adaptive detection. The computational cost and scalability of deep learning models is also not fully discussed. There are no integrated systems for combining traditional and advanced algorithms for increasing robustness. The future work should entail the development of unified, lightweight, and scalable frameworks that can perform real time detection with high performance and adaptability across diverse and dynamic phishing environments. This research focusses on building a model that provides the maximum accuracy and extraction of features from the websites to best achieve the same.

3 Research Methodology

This research is followed by a structured methodology to build an efficient phishing detection system. First, a dataset was acquired that contains features related to phishing websites that is publicly available. Kaggle's "Phishing Dataset for Machine Learning" is built for the identification of phishing web sites via machine learning techniques[1]. It consists of a set of labeled web pages, whose web sites are called 'phishing' or 'legitimate,' along with various features which represent the characteristics of these web sites. There were attributes like URL length and presence of special characters, use of HTTPS, and so on.

[1] https://www.kaggle.com/datasets/shashwatwork/phishing-dataset-for-machine-learning?resource=download

For this research, this dataset formed a basis for experimentation and contained various significant attributes for model training and evaluation. Then the data was preprocessed by changing data types to minimize memory usage and column names to increase interpretability. These steps made the dataset consistent, ready to be analyzed.

The first part of the research was to preprocess the data to get it in the optimal form for its analysis. The features characterizing phishing or legitimate website were available in the dataset. The firsts steps were around identifying and transforming the data types for better memory usage. Feature selection was used in this research to improve model efficiency by utilizing spearman correlation and mutual information scores. A Spearman correlation, a non-parametric measure of rank correlation, was used to find monotonic relations between features and the target variable which helps to select features that have a big predictable power. Moreover, mutual information scores were calculated to measure dependency of each feature and the target variable. This technique quantifies the linear and nonlinear relationships and helps to identify most informative features. The analysis ranked features by applying these methods, which allowed them to prioritize relevant variables, mitigating noise and improving the performance of the models in general.

Splitting the data into training and testing subset was done using random sampling techniques. The 'train_test_split' function from Scikit-learn was used, with 20% of the data split off for testing, and the rest of the data for training. To ensure unbiased representation and have balanced class distributions across both sets, this split was randomized. The model could be generalized to unseen data due to the randomized sampling.

The second phase consisted of training and evaluation of machine learning models. Logistic Regression and Random Forest Classifier were created as two machine learning models. Data was scaled internally for Logistic Regression model to ensure the optimization is stable. Top features were ranked by mutual information scores and used to feed into Random Forest Classifier with parameters for number of estimator and maximum depth tuned to increase the predictive capabilities of random forest classifier.

The performance of the models on the testing subset was evaluated. The effectiveness of the techniques in identifying phishing websites was assessed by computing key metrics like accuracy, precision, recall and F1 score. Thus, these metrics served as a complete evaluation of models based on their capability on correctly predicting each phishing and legitimate website.

4 Design Specification

This thesis aims to design a robust scheme to detect phishing websites using machine learning techniques. The main purpose of this work is to make a prediction model which can make the distinction between legal and phishing websites by using a structured data. To achieve this, the project is executed with the help of Python which contains a rich ecosystem of libraries that

support data processing, visualization and machine learning is taken advantage of. The code was run on Kaggle and used the T4 GPU for faster processing and computation.

Data Preprocessing:

The initial dataset for this project was loaded and analyzed from a CSV file to get the best possible performance. Data preprocessing included converting data types so as to consume less memory. By optimizing it, computational efficiency of data is ensured but not also losing precision.

Feature Analysis:

Analyzing the relationships between features was the critical step in the project. To find significant correlations between features and the target variable (labels), a correlation heatmap function was developed. This allowed us to pinpoint features with high predictive value so that it would be easy to improve the accuracy and efficiency of the machine learning model.

Model Development:

The project aims to create a machine learning algorithm to train and test a predictive model. Data manipulation, analysis and visualization were carried out using Python pandas, numpy, matplotlib, and seaborn libraries. Then with the help of preprocessing the dataset was split into training and testing to check the performance of a model. Only those features that were relevant to the target variable were selected and key features were picked according to the relevance of the selected features to the target variable. The T4 GPU on the Kaggle platform boosted the computational efficiency drastically, making the training and testing of machine learning models much more faster. Kaggle not only enabled use of GPU acceleration, but also made experimentation possible with different algorithms.

Evaluation and Visualization:

One of the important aspects of the project was data and result visualization. To create the nice clear looking charts, correlation heatmap and bar charts showing label distribution, Matplotlib and Seaborn was used. The visualizations helped develop a much deeper understanding of the dataset and therefore the model's performance as well. Standard machine learning metrics, i.e. accuracy, precision, recall and F1 score, were used for model evaluation. Using these metrics, got a good understanding of how the model does in being able to tell fake websites from genuine ones.

5 Implementation

The first step in the development of the model was data preprocessing which is the backbone of the implementation process. This process ensures that the dataset used is clean and optimised

for Machine Learning tasks. First the program uses the Pandas library, to read the dataset to program. Pandas give useful tools for data manipulation. Numerical columns with float64, int64 data types are encoded into float32, int32 respectively to increase the computation efficiency. This step helps to reduce memory usage as it's very important when dealing with big datasets. Furthermore, the column containing target variable CLASS_LABEL is renamed 'labels' for better readability and for clarity in further operations.

Exploratory Data Analysis (EDA) is carried out to try to find patterns and relationships in the data. At this stage, bar plots are used to analyze the distribution of the target labels phishing vs. legitimate) so as to obtain an idea of whether the dataset is balanced or imbalanced. It was found that the dataset was balanced consisting 5000 of both Legitimate and Malicious websites. Using a balanced dataset guarantees fair training of the model, biased one might need oversampling or undersampling. In the next step, the spearman correlation is done where it is possible to find which features are correlated in terms of predicting if a website is phishing or not. This analysis is done by taking 10 columns from the datasets and making a heatmap of the same which helps to understand which all features have a positive or negative impact on whether the website is phishing or not. Afterwards, the mutual info classifier from the sklearn library was used to classify and the features were given a Mutual Info (MI) score according to which their classification was analyzed later on. Hence the MI score helps rank the features based on their importance for the classification of phishing websites. Combining both these analysis types helps to pick the most relevant features for training the model which hence improves computational efficiency and model performance.

The training process was carried out using two models which are Logistic Regression and Random Forest Classifier. The optimal feature set and maximum prediction performance were achieved iteratively by training these models. As a simple, interpretable model, Logistic Regression was chosen as the baseline model. The training was data centric, using feature relevance rather than hyperparameter tuning. Within feature engineering, the features were ranked by the mutual information scores. The model was repeatedly trained with greater amounts of the top ranked features (from 20 to 50). Metrics (precision, recall, F1 score and accuracy) were computed for each iteration.

Since it is robust and able to handle complex data interactions, the Random Forest model was chosen. GPU acceleration was used to speed up the computation when training. The Random Forest model was created with 500 estimators (decision trees), a maximum depth of 32, and the best selection criteria so that the model can learn. GPU based libraries were also used, such as cuml.ensemble to speed up the training. NVIDIA's RAPIDS ecosystem has one of its open-source libraries called cuML. This is designed to supply GPU accelerated machine learning algorithms, letting us compute faster with the help of GPU's parallel processing power. The cuML is used here to speed up training of the Random Forest model in this project. When working with large datasets or when using machine learning models like Random Forest that construct multiple decision trees, the infrastructure backend used by traditional CPU based machine learning libraries (like — scikit learn) can become resource intensive and become very time consuming. Like Logistic Regression, the Random Forest model was also trained

iteratively, adding more top ranked features in every iteration. For each run, the key metrices were calculated: accuracy, precision, recall and F1 score. The performance metrics of the two models were visualized using line plots to help better understand the impact of feature selection and this enabled finding the feature set on which metrics were optimized for each model.

The Final Random Forest Classifier was trained using the top features from the previous training iterations that showed the best performance. These feature set showed a balanced tradeoff between precision, recall and achieving high accuracy in classifying phishing websites.

6 Evaluation

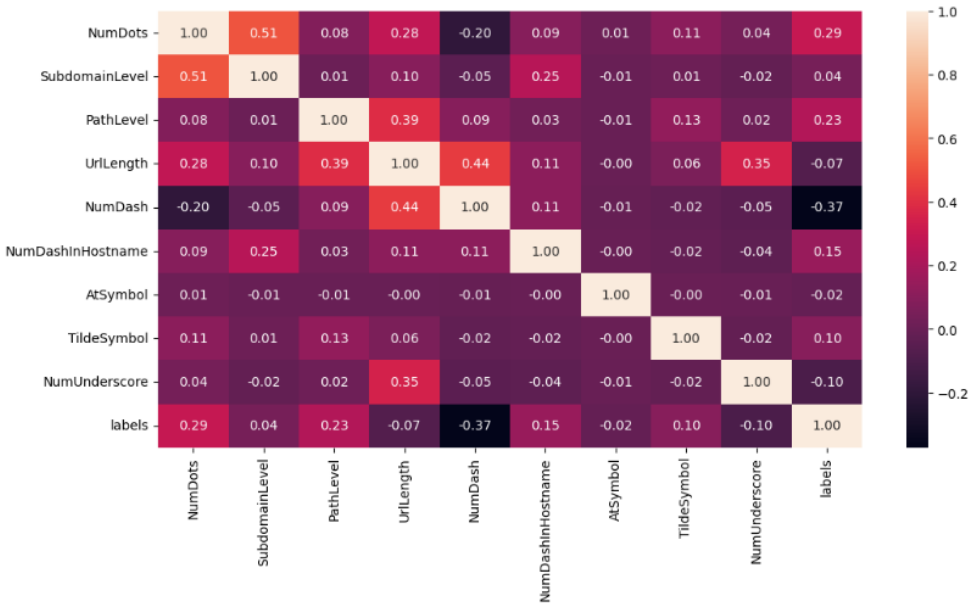
The evaluation metrices used for both Logistic Regression and Random Forest Classifier are accuracy, precision, recall and f1 score. Accuracy is the overall performance measure, that is, it is about the fraction of cases that were correctly classified over all cases. The precision of a model is related to how many of the predicted positives are actually positive, that is, the precision indicates what percentage of its predictions the model correctly predicted. Recall (also known as sensitivity) tells us about the proportion of actual positive instances which are correctly identified, i.e. how good the model is at detecting all actual positive cases. Precision and recall are combined in the harmonic mean of the F1 score, yielding a balanced evaluation which is very effective for imbalanced datasets. Collectively these are detailed metrices to measure the performance of the model.

6.1 Case Study 1- Feature Selection

The Spearman Correlation was used to determine which all features are correlated with each other and its relationship with a target variable which in this case is ‘labels’. The labels variable points to how likely the given set of features contribute to the website being malicious or not. All the columns from the data excluding the ‘id’ column was dropped and a heatmap of the correlation of target variable ‘labels’ with all these other columns are plotted.

- First 10 Columns:

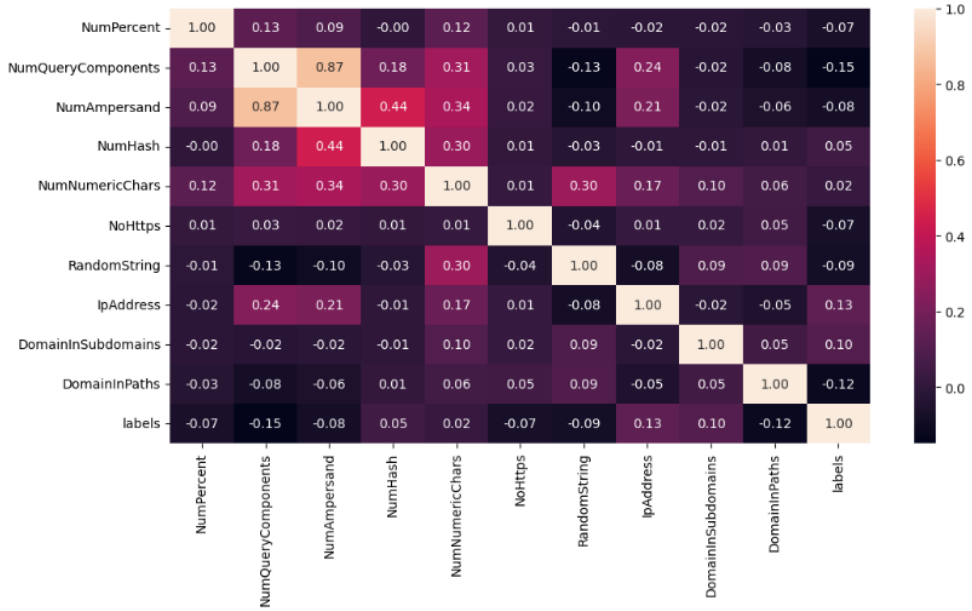
By looking at the first 10 columns against labels, it can be concluded that none of the features have strong correlation with the labels. But it is seen that NumDash has a negative correlation (appx. –0.37) with the labels feature, suggesting that URLs with more dashes are less likely to be associated with malicious URLs (Fig.1).



(Fig.1)

- Columns 10 to 20:

No strong, or even medium, strength correlation features with labels (Fig.2).



(Fig.2)

- Columns 20-30:

There is still no strong correlation feature with labels (Fig.3).

(Fig.3)

- Columns 30-40:

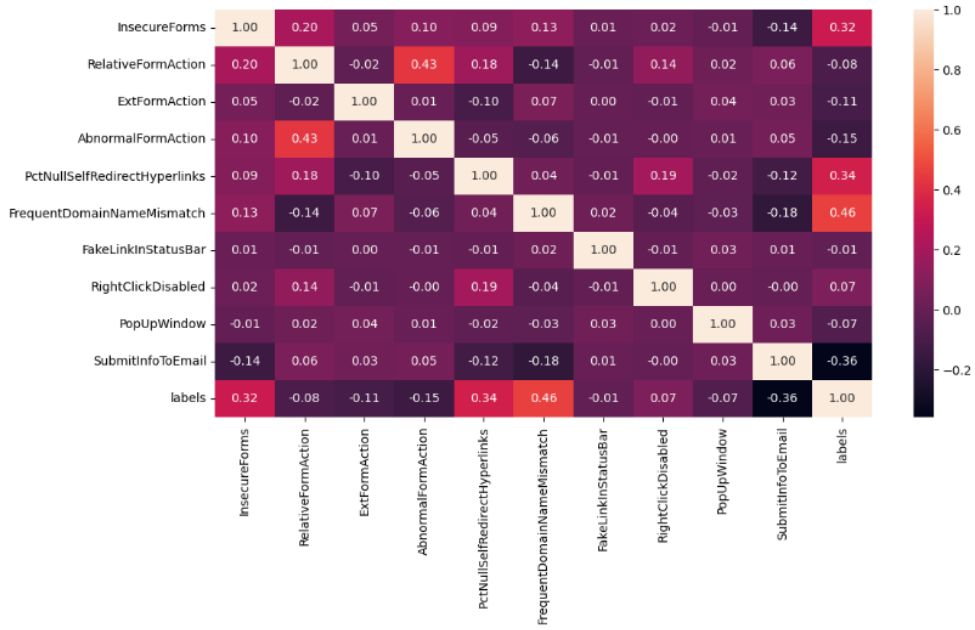
  It is seen there are a few features that are linearly correlated to the target variable 'labels' (Fig.4).

  InsecureForms: Labels have a moderate positive correlation (0.32) to this feature, meaning that certain insecure forms (such as forms that are not using HTTPS) are a significant phishing indicator.

  PctNullSelfRedirectHyperlinks: It has mild positive correlation (0.34) with labels which intimates that a large percentage of null/self redirects can be an indication of phishing URLs.

  FrequentDomainNameMismatch: URLs with frequent mismatches in domains exhibit a positive correlation (0.46) with labels, indicating that phishing is strongly associated with URLs which have mismatches.

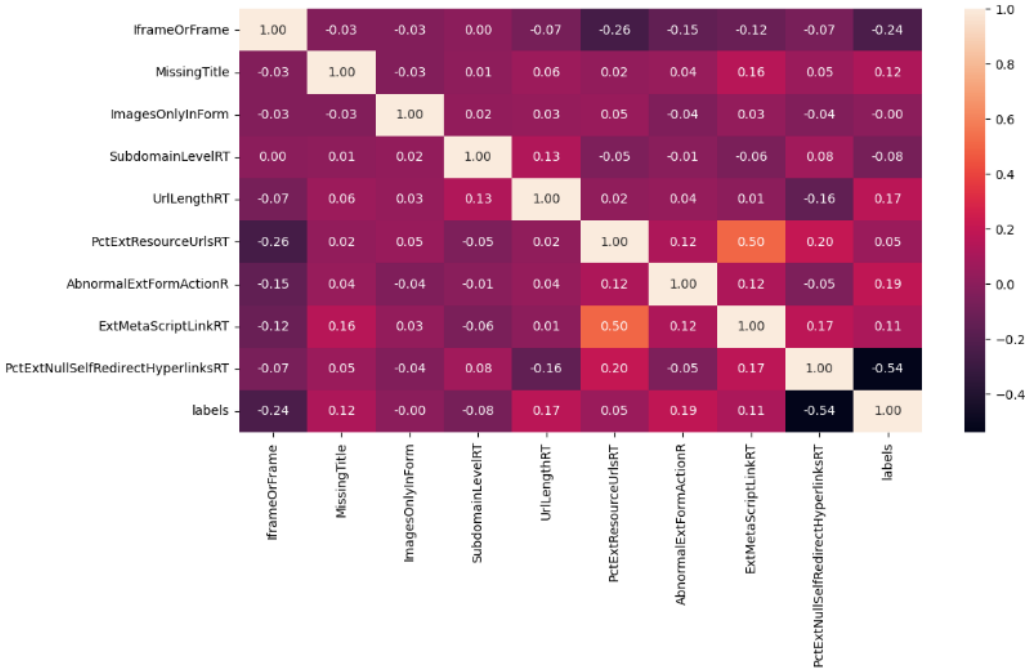
  SubmitInfoToEmail: There is a significant negative correlation (-0.36) to labels meaning that phishing URLs are less likely to use this feature (e.g., submitting data directly to email).



(Fig.4)

- Columns 40-50:

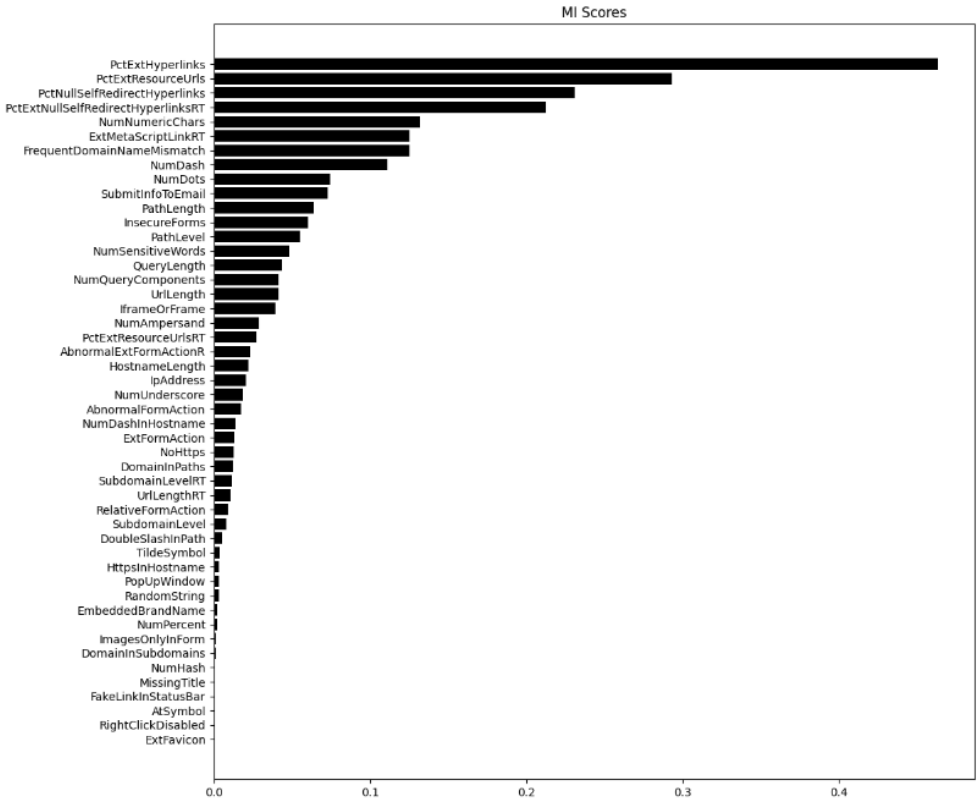
PctExtNullSelfRedirectHyperlinksRT: We find this feature to have a strong negative correlation (-0.54) with labels, meaning higher values in this feature are more strongly correlated with legitimate URLs than phishing URLs (Fig.5).
IframeOrFrame: This feature has a moderate negative correlation (-0.24) with labels, so iframe or frame usage URLs are more likely filtered out or associated with more legitimate URLs (Fig.5).



(Fig.5)

Mutual Info Classifier: Linear correlation between features and labels is found with a mutual info classifier and is stored as mutualinfo_scores. These scores are found to be

slightly different compared to the output obtained from the spearman correlation heatmaps (Fig.6).



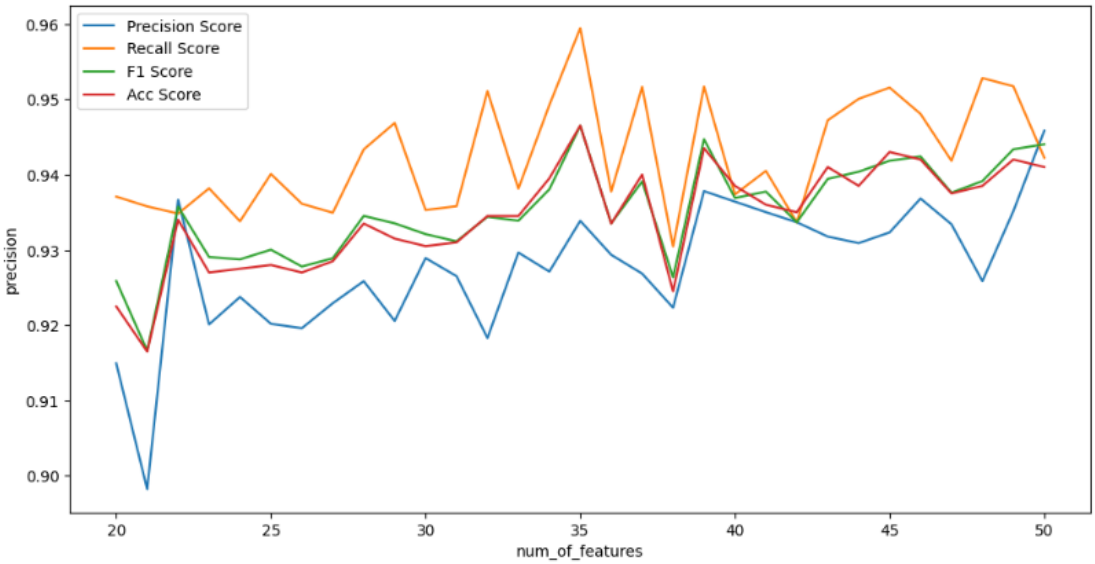
(Fig.6)

The features with high MI scores are more likely to affect whether the website is a malicious one. Those features with the highest MI scores are used as the top n features for training the models which in-turn enhances the efficiency of the model.

6.2 Case Study 2- Logistic Regression

A continuous training process is performed using logistic regression model to find how many features would be required to find the best fitted model without changing the hyper parameters and hence the idea is the data centric training. The method returns number of top N features from the features that scored the highest MI score to be used for training the model and returns all the evaluation metrices.

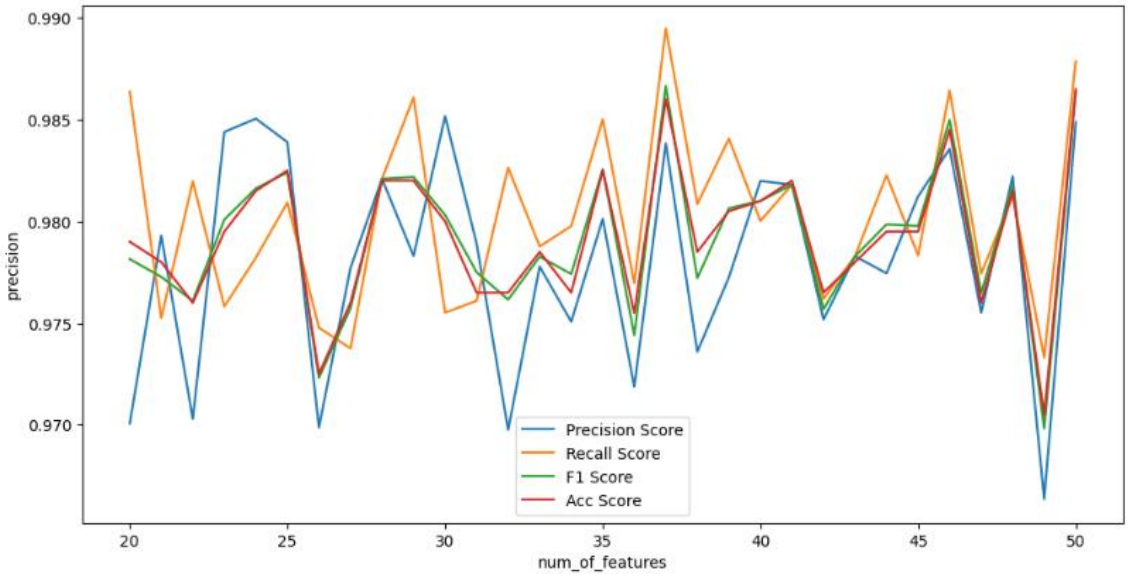
After all these iterations are carried out, the number of features at the highest performance altogether was recorded at 39 features as seen in the graph (Fig.7). The precision for the same was recorded to be 0.937804 which was less than expected as maximum accuracy is one of the objectives in this research. Hence, another model, The Random Forest Classifier was used for training.



(Fig.7)

6.3 Case Study 3 – Random Forest Classifier

The Random Forest Classifier was also trained and the performance of the model in each iteration were recorded. The number of features that showed the highest accuracy was found to be 50 (from Fig.8).



(Fig.8)

6.4 Case Study 4 – Final Random Forest Classifier

The number of features that showed the highest accuracy was 50 (from Fig.9) and the final random forest model was trained using this optimal number of features. The result showed that

the model was capable of predicting with 98% accuracy, precision and recall. This also suggests that this model trained with the provided feature set would best classify phishing websites with high success rate.

	precision	recall	f1-score	support
0	0.98	0.98	0.98	991
1	0.98	0.98	0.98	1009
accuracy			0.98	2000
macro avg	0.98	0.98	0.98	2000
weighted avg	0.98	0.98	0.98	2000

(Fig.9)

6.5 Discussion

The Random Forest Model clearly showed better performance compared to Logistic regression model and hence optimal number of features was selected by training the Random Forest Classifier. Using the feature sets that gave the best accuracy in the iterations, in the same model proved to portray 98% accuracy along with precision and recall for the model which has a high chance of classifying the phishing websites accurately.

7 Conclusion and Future Work

The critical task of phishing website detection was successfully solved using machine learning techniques in this project. Substantial data preprocessing, feature engineering, and training robust models were involved in the implementation in order to achieve an accurate prediction. The first model was built with the help of the Logistic Regression model, serving as a baseline coming up with initial insights, however, the Random Forest Classifier served as the final answer and was used since it outperformed the earlier models. It was shown that the final model has a very high accuracy (98%), and moreover, precision and recall are high; all of this predicts the reliability and effectiveness of the model in identifying the phishing sites. The results show the value of feature selection and iterative exploration in successful modeling. Furthermore, the project showed the value of modern computational tools leveraging the parallel processing abilities of GPUs to accelerate machine learning workflows. The project overall offers a solid base for using machine learning based phishing detection systems in a real-world scenario, by providing a scalable and efficient means to face cyber threats.

Despite this success in the development of a high performing phishing detection model, there are numerous opportunities for this project to be further expanded and its capabilities further enhanced. These improvements could help overcome limits and increase accuracy and offer adaptability to changing cyber threats. A good area to improve is more features. The current model is primarily based on statistical and structural attributes of websites. Future work might

study the inclusion of content-based features e.g. text analysis of website content or metadata extraction to improve its capacity to identify phishing patterns. Moreover, data of users' behavioral, such as users' interactions, history in navigation, bring popular insights to establishing a more robust detection. Also, the inclusion of Deep Learning models like Convolutional Neural Networks (CNN) or Recurrent Neural Networks (RNN) could help in better analyzing the patterns dynamically so that it would handle variety of threats more easily. In addressing these areas, further efforts on the part of future development can increase the capabilities of this system, enabling a more robust, scalable and more effective phishing detection system. This not only improves the model performance but emboldens its role as a key resource to fight the increasing prevalence of phishing attacks.

References

Mahajan, R. and Siddavatam, I. (2018). Phishing Website Detection using Machine Learning Algorithms. International Journal of Computer Applications, 181(23), pp.45–47. doi:https://doi.org/10.5120/ijca2018918026.

Burbela, K. (2023). Model of detection of phishing URLs based on machine learning. [online] Available at: https://www.diva-portal.org/smash/get/diva2:1773760/FULLTEXT02.

Ieee.org. (2024). Available at: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9740763 [Accessed 4 Aug. 2024]..

Bahaghighat, M., Ghasemi, M. and Ozen, F. (2023). A high-accuracy phishing website detection method based on machine learning. Journal of Information Security and Applications, [online] 77, p.103553. doi:https://doi.org/10.1016/j.jisa.2023.103553.

Garje, A., Tanwani, N., Kandale, S., Zope, T. and Gore, S. (2021). Detecting Phishing Websites Using Machine Learning. [online] 9(11), p.243. Available at: https://ijcrt.org/papers/IJCRTI020051.pdf.

Alnemari, S. and Alshammari, M. (2023). Detecting Phishing Domains Using Machine Learning. *Applied sciences*, 13(8), pp.4649–4649. doi:https://doi.org/10.3390/app13084649.

Najwa Altwaijry, Isra Al-Turaiki, Alotaibi, R. and Alakeel, F. (2024). Advancing Phishing Email Detection: A Comparative Study of Deep Learning Models. *Sensors*, 24(7), pp.2077–2077. doi:https://doi.org/10.3390/s24072077.

Dutta, A.K. (2021). Detecting phishing websites using machine learning technique. *PLOS ONE*, [online] 16(10), p.e0258361. doi:https://doi.org/10.1371/journal.pone.0258361.

Said, Y., Alsheikhy, A.A., Lahza, H. and Shawly, T. (2024). Detecting phishing websites through improving convolutional neural networks with Self-Attention mechanism. *Ain Shams Engineering Journal*, [online] p.102643. doi:https://doi.org/10.1016/j.asej.2024.102643.

Sahingoz, O.K., Buber, E., Demir, O. and Diri, B. (2019). Machine learning based phishing detection from URLs. *Expert Systems with Applications*, 117, pp.345–357. doi:https://doi.org/10.1016/j.eswa.2018.09.029.

James, J., L, S. and Thomas, C. (2013). *Detection of phishing URLs using machine learning techniques*. [online] IEEE Xplore. doi:https://doi.org/10.1109/ICCC.2013.6731669.

Pradeepthi, K.V. and Kannan, A. (2014). Performance study of classification techniques for phishing URL detection. *International Conference on Advanced Computing*. doi:https://doi.org/10.1109/icoac.2014.7229761.

Sri Hari Nallamala, Kommu Namitha, Kunchanapalli Raviteja, Kadiyam Sai Sumanth and Jyothi Sri Kota (2024). Phishing URL Detection using Machine Learning. *International Journal for Research in Applied Science and Engineering Technology*, 12(3), pp.1984–1995. doi:https://doi.org/10.22214/ijraset.2024.59261.

Aniket Garje, Namrata Tanwani, Sammed Kandale, Twinkle Zope and Gore, S. (2021a). *IN-DEPTH STUDY OF DETECTION OF PHISHING URLS USING MACHINE LEARNING*. [online] doi:https://doi.org/10.13140/RG.2.2.19399.98729.

D, A. (2019). Phishing Websites Detection using Machine Learning. *International Journal of Recent Technology and Engineering*, 8(2S11), pp.111–114. doi:https://doi.org/10.35940/ijrte.b1018.0982s1119.

Chiew, K.L., Tan, C.L., Wong, K., Yong, K.S.C. and Tiong, W.K. (2019). A new hybrid ensemble feature selection framework for machine learning-based phishing detection system. *Information Sciences*, 484, pp.153–166. doi:https://doi.org/10.1016/j.ins.2019.01.064.