**Feature Extraction Based Ensemble Stacking for Combating Cyber Threat in Phishing** **URLs**

Dasari Siva Krishna1, Assistant Professor, GMRIT, Rajam, India

Tutika Lavanya2,Pydisetti Satyamani Sai2,Renuka Kola2,Sri Samhita Pedaprolu2,SKSadhik Basha2, Tatipalli Harikrishna2,Uppada Manikanta2 GMRIT, Rajam, India

**ABSTRACT**

Cyber threat is a process of exploiting sensitive information by losing the golden triad (Confidentiality, Integrity, and Availability). The threats may include Trojans, Phishing, and DDOS attacks. Phishing works as a social engineering technique where in the malicious attacker attempts in such a way that the appearance to the user would seem legitimate thereby causing the user to fall in a trap which may lead to the compromising of the user’s sensitive information by means of illegitimate URLs or links etc. According to previous records approximately 18% of businesses lost their financial information due to phishing attack. Phishing can exploit data such as credentials, personal data, medical, etc. In this work we proposed a feature extraction-based stacking model where the URL (uniform resource locators) is identified as a legitimate or illegitimate URL. In the process of feature extraction, the features are considered based on hostname\_length, path\_length, fd\_length, tld\_length, count of -, count of @, count of? count of %, count of., count of =, count of http, count of https, count of www, count of digits, count of letters, dir., Ip (v6 or v4). We trained the model with different machine learning algorithms for chosen the best models for stacking the models. We considered the best four machine learning algorithms to be stacked and a meta layer for the final prediction. We compare the metrics such as accuracy, recall, precision, and F1-score with other state of art models.

**Keywords:** Tokenization, Word Embedding, Stacking, Machine Learning, Feature Extraction.

**I - INTRODUCTION**

Phishing is a type of attack where a false website impersonates a reputable website to trick online visitors. After someone visits the false website, the script that is embedded there may take private data such the user name, password, credit card number, bank account number, etc. These days, most financial transactions, payment services, and other activities are carried out online, making them prime targets for phishers. Traditional categorization methods for malicious URL detection similar to regular expression, signature and blacklisting methods are difficult due to massive data., complex patterns that change over time. URL can be utilized to track any website. It is a return address for a website sources. Each URL consists of two primary parts. First up is protocol (HTTP/HTTPS) and second one is resource identifier. The suggested work in this paper explores the evaluation metrics of several Machine Learning classifiers and takes into account the identification of problematic URLs.

The main limitation in the previously published papers is that phishing URLs [2, 8] are not completely identified by the models, and feature selection and other techniques take more time. The Random Forest method [6, 7] performed less accurately compared to others since it took more features into consideration than were necessary to determine whether a URLs [4, 5] was valid or not. Consideration of more features [9, 10] when training the model is the key issue with several of the models in the earlier works. Therefore, the model can be optimized by excluding unimportant features from consideration. The bulk of models achieved good accuracy by using word embedding approaches rather than feature selection. Every single model that has been proposed either uses machine learning [13, 14, 16] or deep learning [12, 15, 17]. Combining both the learning’s with stacking [11,18] will make it more effective to recognize authentic URLs and non-authentic URLs.

*Contribution of the proposed work:*

* Preprocessing of the data is done by making the URLs into tokens by splitting url using some special character (/, //, -,.).
* Features such as hostname length, pathlength, fd\_length, tld\_length, count of -, count of @, count of? count of %, count of., count of =, count of http, count of https, count of www, count of digits, count of letters, dir., Ip (v6 or v4) are extracted to determine each URL type and based on that the URL prediction has done.
* Top 4 best performed machine learning models are given to the ensemble stacking for output predictions.

**II- LITERATURE SURVEY**

A Phishing URL Detection was developed by *M. Sánchez-Paniagua, E. F. Fernández, E. Alegre, W. Al-Nabki, and V. González-Castro [2]* A Real-Case Scenario Through Login URLs produced a new dataset called Phishing Index Login URL (PILU-90K), which is made up of 30K phishing URLs and 60K genuine URLs, comprising login and index websites. Raw words are extracted from the various parts of the URL by segmenting the string using several symbols, including "/," "-," ".," "@," "?"," "?", "&," "=", Eight supervised classifiers are trained and compared to the model, including Light Gradient Boosting Machine, Extreme Gradient Boosting, Adaptive Boosting, Random Forest, Support Vector Machines, k-Nearest Neighbors, Nave Bayes, and Logistic Regression. Accuracy, precision, recall, and f1-score are the performance measures for URL detection. When combined with Term Frequency - Inverse Document Frequency (TF-IDF) feature extraction, the logistic regression model achieves 96.50% accuracy on the recently released login URL dataset.

*Baseline machine learning algorithms:*

*Anand, P., Pais, A.R., and Rao, R.S [6]* using a TWSVM classifier, a heuristic method is used to identify phishing websites. The dataset is derived from Alexa's ranking of the top websites, and the phishing websites are gathered from the Phish Tank website. In contrast to other research, we chose 5500 trustworthy websites at random from the Alexa database, avoiding bias toward highly ranked websites. Moreover, 5500 phishing URLs. The new characteristics are created using TF-IDF words similarity, copyright similarity, title similarity, description similarity, maximum frequency domain similarity, and filename similarity. The machine learning methods Support Vector Machine, Twin Support Vector Machine, and Proximal Support Vector Machine have been used to train and test the dataset. Using a Dell Precision T1700 CPU and 16 GB of RAM, the SVM, PSVM, and TWSVM machine learning algorithms have all been simulated in the MATLAB 12.0 environment. The criteria employed are recall, specificity, precision, f1 score, and accuracy; TWSVM received results of 98.05% accuracy, 97.77% specificity, 98.33% recall, 98.74% precision, and 98.03% f1 score.

A Robust Ensemble Machine Learning Model for Filtering Phishing URLs was proposed by *P. L. Indrasiri, M. N. Halgamuge, and A. Mohammad [7]:* A reliable and acceptable dataset for an ML-based detection technique for URL validity predictions is Expandable Random Gradient Stacked Voting Classifier (ERG-SVC). A dataset including 102400 legitimate URLs and 137375 phishing URLs is utilized to accomplish this study. A feature extraction process is used to determine whether a website is authentic if it has the following characteristics: IP, pagerank0, position (/)>7, non-standard port number, short URL with https in the domain, URL length>75, and domain age= (10 months). Machine learning classifiers includingDT,RF, Boost, AdaBoost, KNN, Gradient Boost, and Logistic Regression were used in this project. The Gradient Boost method outperformed the other algorithms in terms of accuracy, precision, recall, and f1 score. Accuracy 96.71%, precision 96.34%, recall 97.2%, and f1 score 96% were attained.

*Nagendra Areti A model developed by Soma Charan, Yu-Hung Chen, and Jiann-Liang Chen [13]* uses URLs as a dataset to identify phishing websites. Ten features were retrieved from the dataset, which contains 6000 URLs, and were then used to identifywhether a website was phishing. For this study, eight machine learning algorithms were created. According to the performance analysis findings, when compared to other algorithms, the Multilayer Perceptron algorithm had the greatest accuracy (85.41%) and F1 score (85.17%).

*Safa Alrefaai, Ghina zdemir, and Afnan Mohamed [14*] is intended to use machine learning to identify phishing websites. We took data from Kaggle that included 11,430 URLs altogether, 11,430 features, and half of them were phishing sites. The Decision Tree (DT), Random Forest (RF), Boost, Multilayer Perceptron, K-Nearest Neighbors, Naive Bayes, AdaBoost, and Gradient Boosting models were used to train our data, and XG Boost produced the model with the highest accuracy (96.6%).

*Brij B. Gupta, Krishna Yadav, Imran Razzak, Konstantinos Psannis, Arcangelo Castiglione, and Xiaojun Chang[16]* used the ISCXURL-2016 dataset to test 11964 instances of legal and phishing URLs. The characteristics were taken from the URLs and subjected to training against various machine learning classifiers, with the Random Forest technique yielding the highest accuracy of 98.07%.

*Jalil, S., Usman, M., and Fong[19]* suggested a technique based on URL that employs the whole URL, protocol scheme, hostname, path region of the URL, entropy feature, suspicious terms, and brand name matching using TF-IDF technique. Eight different machine learning classifiers were used in the trials, which were conducted on six different datasets. Random Forest significantly outperformed the other classifiers in terms of accuracy across all datasets. On the Kaggle datasets, the suggested framework with just 30 features produced superior accuracy results of 96.25% and 94.65%. The comparison findings demonstrate that the suggested model outperformed the existing methods with accuracy of 92.2%, 91.63%, 94.80, and 96.85% on benchmark datasets.

*K.S.R.C. Murthy, T. Bhattacharya, and N. Rajagopalan [20]* machine learning techniques for feature extraction-based phishing url detection. The objective of this study is to recognize phishing URLs using address bar and behavioral data. The decision tree method, which the author took into consideration and cross-validated on a few machine learning techniques, is found to be effective in categorizing phishing URLs according to the discovered features. 9581 data points and 41 characteristics make up the dataset used in this study. Finally, author was able to locate phishing URLs with 97% accuracy, which is comparatively greater accuracy than other detection algorithms.

*Deep learning algorithms:*

A model combining long-term recurrent convolutional and graph convolutional networks to detect phishing sites was proposed by *S. Ariyadasa, S. Fernando, and S. Fernando [1]* using HTML and URL. In the current inquiry and experiment, three datasets were used. One of them is exclusive to our study, whereas the other two may be found in publicly available databases. For ease of use, the datasets were given the designations Dataset A, Dataset B, and Benchmark Dataset. The experiment explored training, testing, and validation on various scales using these three datasets. There are 45003 reliable URLs and 50000 real URLs altogether. To Extract Features examines three main feature sets for existing anti-phishing technology. These are characteristics that are external, content-based, and URL-based. URL-based features are lexical features that may be extracted directly from the URL, as opposed to content-based features, which are obtained directly from HTML text. Random forest, KNN, naive bayes, and deep learning models like LSTM and GRE are some of the machine learning algorithms that have been employed. By comparing the findings, it was found that LSTM outperformed other algorithms. Accuracy, precision, recall, and f1-score for URL detection are the performance measures. Accuracy, precision, recall, and f1 score were attained at 96.42%, 96.40%, and 96.44%, respectively.

*Aljofey, A., Jiang, Q., Qu, Q., Huang, M., and Niyigena, J.-P[12]* authors does not call for retrieving the content of the target website or utilizing any third-party services. Without having any prior knowledge of phishing, it collects data and sequential patterns of URL strings, using the sequential pattern features for quick classification of the actual URL. Comparisons between numerous classical machine learning models and deep learning models are offered for evaluations utilizing a range of feature sets, including hand-crafted, character embedding, character level TF-IDF, and character level count vectors features. The suggested model, according to the experiments, had accuracy of 95.02% on the dataset and 98.58%, 95.46%, and 95.22% on benchmark datasets.

To identify and categorize phishing websites*, Salloum, Gaber, Vadera, and Shaalan* *[15]* offered a method to develop classification models utilizing features retrieved from websites. The author employed two datasets of 58,645 and 88,647 URLs that were classified as "Phishing" or "Legitimate" to train the system. There are several different machine learning models that are examined, including "XGBOOST, Support Vector Machine (SVM), Random Forest (RF), k-nearest Neighbour (KNN), Artificial neural network (ANN), Logistic Regression (LR), Decision tree (DT), and Gaussian naive Bayes (NB)" classifiers. In tests, ANN demonstrated the best performance with a 97.63% accuracy rate for phishing URL detection.

A hybrid rule-based system was proposed by *Youness Mourtaji, Mohammed Bouhorma, Daniyal Alghazzawi, Ghadah Aldabbagh, and Abdullah Alghamdi* *[17]* to incorporate 37 features taken from six different methods, including the black listed method, the lexical and host method, the content method, the identity method, the identity similarity method, the visual similarity method, and the behavioral method. Later Different deep learning and machine learning models, including MLP and CNN (Convolution Neural Network), are utilized for training, in addition to machine learning models like CART, SVM, and KNN. The study's conclusions showed that the approach was successful in analyzing URL stress from various angles, supporting the model's validity. However, deep learning was able to achieve the highest accuracy level with the provided values of 97.945 for the CNN model and 93.216 for the MLP model.

*Stacking approaches:*

*Jay Chadawar and Anurag Pandey[3]* Detection of Phishing URLs Using Hybrid Ensemble Model, A total of 20,000 normal and malicious URL combinations make up the dataset under consideration. The dataset's URLs are sent to a variety of features, which, depending on the circumstances, return 0 or 1. The returned values are then tabulated saved in a csv file. Training and testing data are separated from the dataset in varying ratios. The dataset is applied to the imported classifiers, and the corresponding accuracies are determined. To create weak learners in this work, we define a set of models a variable number of times. Finally, the Max Voting Classifier approach is employed, and the ensemble model's final class prediction will be the one that has been primarily predicted by the weak learners. This hybrid model has an accuracy of 85.37%.

A malicious URL detection method was presented by *Maheshwari, Shantanu, Janet, and Kumar[4]* utilizing A 450000 URLs public dataset from Kaggle is the data's source. The top classifier locates harmful URLs on the publicly accessible phishing website. The collection includes labels for the URLs that indicate whether they are malicious or benign. There are 104438 harmful URLs and 104438 benign URLs. ​Pre-processing includes management of missing data, extrapolation of new features, normalization, encoding of categorical values, and standardization of values. The number of layers and kernels affect the CNN's performance in general. Thesklearn Python Library is used to train the model on 80% of the data using several machine learning techniques, such as Nave Bayes, Nave Regression, Stochastic Gradient Descent, Random Forest, Support Vector Machine, K-Nearest Neighbors, and Decision Tree. The metrics recall, precision, support, and f1 were used to train and assess a variety of models. The Random Forest model produced the best results. The recall is 91.2%, the recall score is 90%, the accuracy is 97%, and the precision is 91%.

Machine learning techniques, *M. H. Alkawaz, S. J. Steven, A. I. Hajamydeen, and R. Ramli[5]* offered a Comprehensive Survey on Identifying and Analyzing Phishing Websites. Phish Alert fared better on an experimental dataset of 500 phishing sites and 500 real sites. While the phishing URL was received using phish tank, the real URLs were retrieved through the stuff gate server. Data tokenizing techniques like stemming and lemmatization have been utilized for pre-processing. Decision Tree, Random Forest, and Support Vector Machine are the machine learning models that were applied in this instance. The model's accuracy has altered based on the splitting ratio. Accuracy, a false positive rate, and a false negative rate are the measures used for assessment. The accuracy is 97.14 and false positive rate 2.62%, false negative rate is 3.14 for Random Forest model training and testing​.

*A. El Aassal, S. Baki, A. Das, and R. M. Verma[8]* utilizing 10,000 emails that included URLs in total. It retrieved 8,433 emails from the Nazario phishing email collection in addition to 1,048 emails from its recently released 2015 to 2017 emails. The 1,019 Spam Assassin emails were included. Utilizing Information Gain, the Gini Index, the chi-square test, and Recursive Feature Elimination, the features are processed and ranked. SVM, Random Forest, Decision Tree, Gaussian & Multinomial Naive Bayes, Logistic Regression, K Nearest Neighbors, Boosting, Bagging, Online Learning, Deep Neural Networks, Imbalanced learning, and Hellinger Distance Decision Tree are the models that are employed in this model. Accuracy, Recall, F1-score, Precision, Geometric Mean, Balanced Detection Rate, and Matthew's Correlation Coefficient are the performance indicators for identifying URLs. Gaussian Naive Bayes among trained models had an accuracy of 94.7%, F1 score of 94.36%, GMean of 94.48%, and BDR of 17.35%.

A multi-layered stacked ensemble learning technique was presented by *Kalabarige, Lakshmana, Rao, Routhu, Abraham, Ajith, and Gabralla, Lubna.[11].* It consists of estimators at several layers, with predictions from the current layer serving as input to the subsequent layer. The dataset utilized has 10052 records, each with a unique URL and information about how valid or not they are. The suggested model Random Forest is tested using data from Mendeley 2018 (D2), Mendeley 2020 (D1), and UCI (D1) (D3, D4). With the D1 dataset, the suggested model has a detection rate of 97.76% and a D2 dataset accuracy of 98.9%. Finally, D3 and D4 are used to test the approach, with accuracy of 96.79% and 98.43%, respectively.

*Feature learning approaches:*

According to the SPWalk: Similar Property Oriented Feature Learning for Phishing Detection proposed by *X. Liu and J. Fu [9] .T*here are 0.5 million harmful URLs from Phish Tank and Open Phish and 1 million reliable URLs from Alexa and Dmoz. These annotated URLs were used in the calculation of the URL quality score. The TF-IDF will be used to separate the numeric data from the word data once the URLs have been separated into three training layers, L0, L1, and L2. Network embedding models like Deep Walk and Node2vec as well as conventional phishing detection models like URLPatternMining and Website Logo are employed as embedding strategies in this model. the evaluation of the SPWalk, Deep Walk, and Node2vec findings. In this case, the SPWalk consistently outperforms both Deep Walk and Node2vec.Precision, which is 95%, has been utilized as the parameter to assess performance.

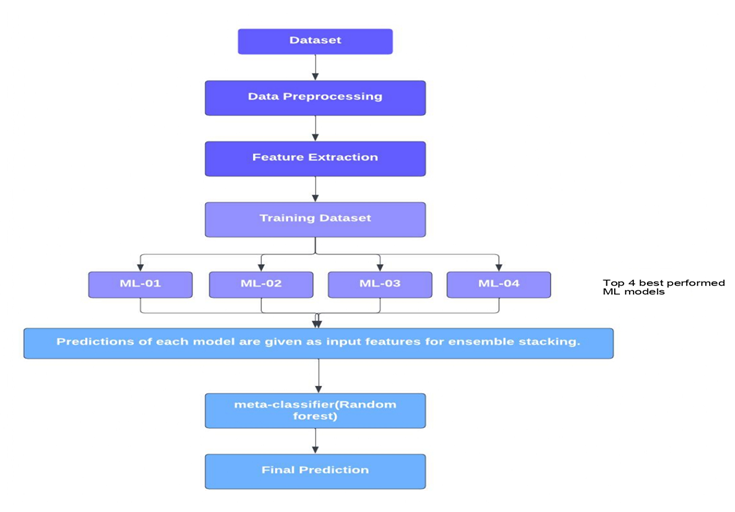
Stop-Phish, an intelligent phishing detection system utilizing feature selection ensemble, was proposed by *Ramana, A., Rao, K., and Rao, Routhu[10]*. The 11055 webpages that make up the UCI repository's dataset for the experiment are used. The remaining 4898 are phishing websites, leaving 6157 legal websites. To find the best classifier, we applied a variety of machine learning methods and created an ensemble model using the Random Forest, Decision tree, and Boost algorithms. According to our experimental research, we were able to detect phishing attempts with an accuracy of 97.51% using a dataset from UC Irvine (Dataset 1) and 98.45% using a dataset from Mendeley (Dataset 2). Additionally, the proposed model significantly outperformed baseline models.

Transform learning:

*P. Maneriker, J. W. Stokes, E. G. Lazo, D. Carutasu, F. Tajaddodianfar, and A. Gururajan[18]* proposed a model employing transformers. The author compared these models to refined BERT and RoBERTa models while also considering extra domain-specific pre-training tasks and standard masked language modelling. Combining the information from these studies, the author derivedURLTran, which outperforms existing deep learning-based algorithms in terms of phishing URL identification over a wide range of extremely low false positive rates (FPRs)? When compared to the next best baseline at an FPR of 0.01%, URLTran, for instance, yields a true positive rate (TPR) of 86.80% as opposed to 71.20%, representing a relative improvement of almost 21.9%.

**III- METHODOLOGY**

Using certain pre-processing methods from earlier phishing models, the dataset [5][19] is prepared. The features associated with the URL will then be extracted. Then, to identify phishing websites based on URLs, the features will be learned using a variety of machine learning models, among which the most accurate models will be trained using the data ensemble stacking model [3,11]. The major parts of this work were divided into a number of modules, including pre-processing, splitting, training, vectorization [2,6,9], and feature extraction [7], as well as an ensemble stacking [3] model for the detection of phishing URLs, as shown in figure [1].

****

*Figure 1: Proposed work data flow diagram*

***3.1 Dataset***

The source of the data is a 4,50,000 URLs public dataset from Kaggle. On the open phish website, dangerous URLs are found using the best classifier. The dataset includes URLs and labels indicating whether they are benign or malicious. 1,04,438 URLs are malicious and the other are benign [5][19]. If result is 0 then the URL is legitimate else it is phishing.

|  |  |  |
| --- | --- | --- |
| **url** | **label** | **result** |
| https://www.google.com | benign | 0 |
| https://www.youtube.com | benign | 0 |
| http://taxcom-online.ru/about/bil.htm | malicious | 1 |
| http://tleg.org/jerry | malicious | 1 |
| http://pacificventurebd.com/biz/ | malicious | 1 |
| https://www.logostelos.info/ | benign | 0 |

*Table 1: Dataset samples*

***3.2 Data Pre-processing***

 In data pre-processing, tokenization [5] is a strategy used to split the URL into words based on some unique characters which are analysed by using the previous papers. The pre-processing looks at the presence of unique characters (//, /, -,.,,,) [2]. Every URL will consist of its own tokens. In the dataset the column label is not required because to predict whether URL is legitimate or not, we require two cases 0, 1. So label column will be deleted and if there are any records which are not having both url and result it will be dropped from the dataset.

***3.3 Feature Engineering***

*3.3.1 Vectorization*

In our model we are utilizing two vectorization strategies such as TF-IDF [2, 6, 9, 12] and Word2vec [12, 18].

Using TF-IDF the token from pre-processing step will be converted to numbers which will be valid to train with machine learning algorithms. The url tokens in the splitted dataset I.e., training and testing sets will be converted to vectors.

*tfidf(t,d,D)=tf(t,d)⋅idf(t,D)*

*tf(t,d)=∑t∈dt|d|*

*idf(t,D)=1+log(1+|D|1+∑t∈Dt)*

Using Word2vec each url token in the training and testing set will be given to model.wv[token] which will give corresponding vector to every token. Vectors corresponding to every url will be appended into two lists which will be trained further using Machine learning models.

*3.3.2 Feature extraction*

In the proposed model we should work with more features without relying on word2vec and TF-IDF there are other ways to convert features [13,16,17,19,20] into numbers, such as by using Matlab and feature extraction. The feature extraction logic [1] is used in this proposed model where the features that are been extracted are hostname length: Host name is the string containing domain name of the URL. Each element of a host name must be from 1 to 63 characters long and the entire host name, including the dots, can be at most 253 characters long, path\_length: path length is the hop count which is intermediatory nodes between the source and destination, Tld\_length: It is used in splitting host into domain and sub domain components. It can be in the range of 2 to 63 characters long, fd\_length, count of -, count of @, count of? count of %, count of., count of =, count of http, count of https, count of www, count of digits, count of letters, Dir: each web page has a URL directory where the URLs are placed hierarchically. It shows the directory in which it is present, and in which format the url Ip is following such as Ip (v6 or v4) [7]. Total of 20 features are collected from each URL but out of which 17 features will be useful for further predictions.

***3.4 Word2vec based Ensemble stacking model***

In training phase, the training vector list will be given as input to the baseline machine learning algorithms such as Naïve Bayes, TWSVM, KNN, SVM, LR, RF, Decision Tree. While testing, the top 4 models which gave high accuracy such as LR, TWSVM, KNN, Decision Tree are given for ensemble stacking model for the output prediction at level-0 base line and at meta layer RF are tested for the prediction of phishing URL. The performance of each model individually tested which shown in table [2].

***3.4 TF-IDF based Ensemble stacking model***

In training phase, the training numerical will be given as input to the baseline machine learning algorithms such as Naïve Bayes, TWSVM, KNN, SVM, LR, RF, Decision Tree. While testing, the top 4 models which gave high accuracy such as LR, TWSVM, Naïve bayes, Decision Tree are given for ensemble stacking model for the output prediction at level-0 base line and in meta layer RF are tested for the prediction of phishing URL. The performance of each model individually tested which shown in table [3].

***3.5 Feature extraction-based Ensemble stacking model***

In Feature extraction-based stacking model, we converted the URLs into 17 features in which each feature having its own representative in identification of phishing URLs. These features are trained with different base line ML models such as naïve Bayes, TWSVM, KNN, SVM, LR, RF, Decision Tree. While testing, the top 4 models which gave high accuracy such as LR, TWSVM, KNN, Naïve bayes are given for ensemble stacking model for the output prediction at level-0 base line and in meta layer RF are tested for the prediction of phishing URL. This model ended up with better accuracy than the other approaches which shown in table [4].

**IV - EXPERIMENTAL SETUP**

In evaluation of the proposed work, we first considered the vectorization techniques such as TF-IDF and word2vec. We tested the performance measures with baseline machine learning model in which ML models are under performed. We also tested the baseline model with ensemble stacking approaches this improves the accuracy but precision, recall and f1-score having some deviation in the performance which shown in table [2,3].

In the proposed model, we extracted 17 features using feature extraction techniques and also, we tested the performance with the baseline machine learning in which we got the better performance and stable performance other matrices which shown inn table [4].

***4.1 Evaluation measures:***

To evaluate the performance of the proposed models, standard metrics are used for classification tasks, such as Accuracy, Precision, Recall and F1-Score.

**Accuracy**: It is the number of correctly predicted divided by the total number samples

**Precision:** It is the proportion of positive predictions that are truly positives.

**Recall**: It is the proportion of actual Positives that are correctly classified.

**F1-Score**: It is the harmonic mean of precision and recall.

*4.2 Performance of ML models with word2vec:*

*Table 2: Performance of different machine algorithms with Word2ve*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Algorithm** | **Accuracy** | **Recall** | **Precision** | **F1 score** |
| Naïve Bayes | 66.22 | 34.12 | 47.46 | 39.21 |
| TWSVM | 81.67 | 77.45 | 67.98 | 72.34 |
| KNN | 81.33 | 74.22 | 70.50 | 72.13 |
| SVM | 78.92 | 65.21 | 68.56 | 67.55 |
| Logistic Regression | **84.25** | 63.45 | **84.20** | 72.34 |
| Decision Tree | 82.33 | **77.56** | 73.22 | **74.00** |
| Stacking (RF) | **84.50** | **60.23** | **79.23** | **83.98** |

***Evaluation -1:*** The model is processed into training and testing and the vectors are trained using above algorithms such as Naïve Bayes, TWSVM, KNN, SVM, LR, RF. We tested the word2vec training set with ML models in which accuracy was improved for ensemble stacking but when consider with other parameters such as recall, precision and f1-score the performance was improved in decision tree, logistic regression and decision tree respectively. In the performance of word2vec with baseline ML models are underperformed with proposed models.

*4.3 Performance of ML models with TF-IDF:*

*Table 3: Performance of different machine algorithms with TF-IDF*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Algorithms** | **Accuracy** | **Recall** | **Precision** | **F1 score** |
| Naïve Bayes | 90.67 | 63.90 | 84.09 | 72.34 |
| TWSVM | 87.07 | 45.99 | 72.39 | 85.67 |
| KNN | 85.06 | 34.03 | 47.62 | 39.75 |
| SVM | 87.05 | **78.90** | 78.34 | **86.55** |
| Logisitic Regression | **91.56** | 74.09 | 70.19 | 72.33 |
| Decision Tree | 89.89 | 33.00 | 80.02 | 81.02 |
| Stacking (RF) | 87.08 | 45.93 | **86.56** | 82.34 |

***Evaluation -2:*** We tested the dataset with different ML models using TD-IDF vectorization. The model logistic regression given 91.56% accuracy which was better than the word2vec model. The other matrices such as precision, recall and F1-score also improved compared with word2vec model. We also tested the performance with ensemble stacking approaches which is not stable with other ML models.

*4.4 Performance of ensemble stacking with ML models:*

*Table 4: Performance of different machine algorithms with proposed method*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Algorithm** | **Accuracy** | **Recall** | **Precision** | **F1 score** |
| Naive Bayes | 97.21 | 97.17 | 97.28 | 97.22 |
| Twin SVM | **98.65** | **98.14** | **98.65** | **98.39** |
| KNN | 96.49 | 95.00 | 95.40 | 96.00 |
| SVM | 95.64 | 95.00 | 95.00 | 95.97 |
| Logistic Regression | 96.56 | 96.50 | 96.50 | 96.25 |
| Decision Tree | 95.60 | 95.99 | 95.11 | 95.03 |
| **Proposed work Stacking (RF)** | **99.74** | **99.55** | **99.69** | **99.60** |

***Evaluation -3:*** In this evaluation, we tested the model performance with our proposed feature based stacking ensemble. We chosen the four best performed stacking models for level-0. In metalayer we chosen the random forest at level-1 for better prediction of the model. The performance matrices also stable for each machine learning where Twin SVM performed better than the other ML models. The proposed ensemble stacking with feature based approach performed better than the other ML models shown in table [3].

*4. 5 Comparison with existing work:*

The proposed model was compared with existing literature in which Rao, R.S., Pais et al.[6] authors proposed Twin SVM which performed 98.05% accuracy. M. H. Alkawaz et al.[5] authors proposed Ml models which performed 96% accuracy and A. El Aassal et al.[8] authors proposed ML based techniques which performed 99%. The proposed work significantly improves with existing works. This ensemble stacking work results in high accuracy with other state of art models.

*Table 3: Comparison between proposed and existing works*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Authors** | **Algorithm** | **Accuracy** | **Recall** | **Precision** | **F1 score** |
| M. H. Alkawaz et al.[5] | Decision Tree | 96.80 | - | - | - |
| Random Forest | 96.84 | - | - | - |
| SVM | 96.40 | - | - | - |
| Rao, R.S., Pais et al.[6] | TWSVM | 98.05 | 98.33 | 97.74 | 98.03 |
| SVM | 97.73 | 97.86 | 97.60 | 97.73 |
| PSVM | 95.03 | 95.49 | 94.67 | 95.08 |
| A. El Aassal et al.[8] | AutoSk | 97.00 | 90.02 | 92.46 | 96.53 |
| Boosting | 97.70 | 99.00 | 98.01 | 98.00 |
| SVM | **99.01** | **99.00** | **98.02** | **89.01** |
| **Proposed work Stacking (RF)** | | **99.74** | **99.55** | **99.69** | **99.60** |

**6. Conclusion:** Phishing detection technologies are essential for ensuring users have a safe online experience, preventing online fraud, preventing personal information. In this work, we made a new attempt that conversion phishing URLs into fixed number of features. Based on the extracted features the ensemble stacking model performed better than the existing models. The proposed work tested with mobile application we considered some set of URLs where it was identified the phishing the URL. The purpose of this work is to research and evaluate earlier efforts to determine the best classification scheme for detection of phishing URLs.

**7. Limitations and future work:** The suggested model is only applicable to current URLs. Future changes to the structure properties of URLs used to train the model, such as the emergence of new IP addresses and top-level domains (tlds), may occur. The URL patterns and extensions, such as shortening services, may change in the future, and these issues will need to be addressed.

**References**

1. S. Ariyadasa, S. Fernando and S. Fernando, "Combining Long-Term Recurrent Convolutional and Graph Convolutional Networks to Detect Phishing Sites Using URL and HTML," in IEEE Access, vol. 10, pp. 82355-82375, 2022, Doi: 10.1109/ACCESS.2022.3196018.
2. M. Sánchez-Paniagua, E. F. Fernández, E. Alegre, W. Al-Nabki and V. González-Castro, "Phishing URL Detection: A Real-Case Scenario Through Login URLs," in IEEE Access, vol. 10, pp. 42949-42960, 2022, Doi: 10.1109/ACCESS.2022.3168681.
3. Anurag Pandey, Jay Chadawar, 2022, Phishing URL Detection using Hybrid Ensemble Model, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 11, Issue 04 (April 2022).
4. Maheshwari, Shantanu & Janet, B & Kumar, R. (2021). Malicious URL Detection: A Comparative Study. 1147-1151. 10.1109/ICAIS50930.2021.9396014. ​
5. M. H. Alkawaz, S. J. Steven, A. I. Hajamydeen and R. Ramli, "A Comprehensive Survey on Identification and Analysis of Phishing Website based on Machine Learning Methods," 2021 IEEE 11th IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE), 2021, pp. 82-87, Doi: 10.1109/ISCAIE51753.2021.9431794.
6. Rao, R.S., Pais, A.R. & Anand, P. A heuristic technique to detect phishing websites using TWSVM classifier. Neural Computer & Application 33, 5733–5752 (2021), https://doi.org/10.1007/s00521-020-05354-z.​
7. P. L. Indrasiri, M. N. Halgamuge and A. Mohammad, "Robust Ensemble Machine Learning Model for Filtering Phishing URLs: Expandable Random Gradient Stacked Voting Classifier (ERG-SVC)," in IEEE Access, vol. 9, pp. 150142-150161, 2021, Doi: 10.1109/ACCESS.2021.3124628.
8. A. El Aassal, S. Baki, A. Das and R. M. Verma, "An In-Depth Benchmarking and Evaluation of Phishing Detection Research for Security Needs," in IEEE Access, vol. 8, pp. 22170-22192, 2020, Doi: 10.1109/ACCESS.2020.2969780.
9. X. Liu and J. Fu, "SPWalk: Similar Property Oriented Feature Learning for Phishing Detection," in IEEE Access, vol. 8, pp. 87031-87045, 2020, Doi: 10.1109/ACCESS.2020.2992381.
10. Ramana, A. & Rao, K. & Rao, Routhu. (2021). Stop-Phish: an intelligent phishing detection method using feature selection ensemble. Social Network Analysis and Mining. 11. 10.1007/s13278-021-00829-w.
11. Kalabarige, Lakshmana & Rao, Routhu & Abraham, Ajith &Gabralla, Lubna. (2022). MLSELM:Multi-layer Stacked Ensemble Learning Model to detect phishing websites. IEEE Access. 10. 1-1. 10.1109/ACCESS.2022.3194672.
12. Aljofey, A.; Jiang, Q.; Qu, Q.; Huang, M.; Niyigena, J.-P. An Effective Phishing Detection Model Based on Character Level Convolutional Neural Network from URL. *Electronics*2020, *9*, 1514. https://doi.org/10.3390/electronics9091514.
13. Areti Nagendra Soma Charan, Yu-Hung Chen, Jiann-Liang Chen, "Phishing Websites Detection using Machine Learning with URL Analysis", 2022 IEEE World Conference on Applied Intelligence and Computing (AIC), pp.808-812, 2022.
14. Safa Alrefaai, Ghina Özdemir, Afnan Mohamed, "Detecting Phishing Websites Using Machine Learning", 2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), pp.1-6, 2022.
15. Salloum, S., Gaber, T., Vadera, S., Shaalan, K. (2021). Phishing Website Detection from URLs Using Classical Machine Learning ANN (Artificial Neural Network) Model. In: Garcia-Alfaro, J., Li, S., Poovendran, R., Debar, H., Yung, M. (eds) Security and Privacy in Communication Networks. SecureComm 2021. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 399. Springer, Cham. https://doi.org/10.1007/978-3-030-90022-9\_28.
16. Brij B. Gupta, Krishna Yadav, Imran Razzak, Konstantinos Psannis, Arcangelo Castiglione, Xiaojun Chang,A novel approach for phishing URLs detection using lexical based machine learning in a real-time environment, Computer Communications, https://doi.org/10.1016/j.comcom.2021.04.023.
17. Youness Mourtaji, Mohammed Bouhorma, Daniyal Alghazzawi, Ghadah Aldabbagh, Abdullah Alghamdi, "Hybrid Rule-Based Solution for Phishing URL Detection Using Convolutional Neural Network", Wireless Communications and Mobile Computing, vol. 2021, Article ID 8241104, 24 pages, 2021. https://doi.org/10.1155/2021/8241104.
18. P. Maneriker, J. W. Stokes, E. G. Lazo, D. Carutasu, F. Tajaddodianfar and A. Gururajan, "URLTran: Improving Phishing URL Detection Using Transformers," MILCOM 2021 - 2021 IEEE Military Communications Conference (MILCOM), 2021, pp. 197-204, doi: 10.1109/MILCOM52596.2021.9653028.
19. Jalil, S., Usman, M. & Fong, A. Highly accurate phishing URL detection based on machine learning. J Ambient Intell Human Comput (2022). https://doi.org/10.1007/s12652-022-04426-3.
20. Murthy, K.S.R.C., Bhattacharya, T., Rajagopalan, N. (2022). Feature Extraction-Based Phishing URL Detection Using Machine Learning Techniques. In: Satyanarayana, C., Samanta, D., Gao, XZ., Kapoor, R.K. (eds) High Performance Computing and Networking. Lecture Notes in Electrical Engineering, vol 853. Springer, Singapore. https://doi.org/10.1007/978-981-16-9885-9\_14.