IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 14, No. 1, February 2025, pp. 358~367
ISSN: 2252-8938, DOI: 10.11591/ijai.v14.i1.pp358-367

A multi-algorithm approach for phishing uniform resource locator's detection

Devi Sree Guddanti, Rupa Chiramdasu, Mahalakshmi Gayathri Uppuluri
Department of Computer Science and Engineering, Velagapudi Ramakrishna Siddhartha Engineering College, Vijayawada, India

Article Info

Article history:

Received Apr 16, 2024
Revised Jul 30, 2024
Accepted Aug 30, 2024

Keywords:

AdaBoost
CatBoost
Cybersecurity threats
Multi layer perceptron
Random forest
Uniform resource locators
phishing

ABSTRACT

Nowadays, the internet is used to organise a wide range of cybersecurity risks. Threats to cybersecurity include a broad spectrum of malevolent actions and possible hazards that affect data, networks, and digital systems. Cybersecurity dangers that are commonly encountered are distributed denial-of-service (DDoS) attacks, phishing, and malware. Phishing attempts frequently use text messages, email, and uniform resource locators (URLs) to target specific people while impersonating trustworthy sources in an effort to trick the victim. Consequently, machine learning plays a critical role in stopping cybercrimes, especially those that involve phishing assaults. The suggested model is based on a well constructed dataset that has been enhanced with 32 features. By combining the features of several machine learning methods, such as random forest, CatBoost, AdaBoost, and multilayer perceptron, the suggested model greatly increases the precision of phishing URL detection. Evaluation indicators that highlight the model's effectiveness in defending against cyber threats include precision, recall, accuracy, and F1-score. These metrics also highlight the urgent need for proactive cybersecurity measures.

Corresponding Author:

Devi Sree Guddanti
Department of Computer Science and Engineering, Velagapudi Ramakrishna Siddhartha Engineering College
Vijayawada 520007, India
Email: devisree1293@gmail.com

1. INTRODUCTION

Cybersecurity is seriously threatened by phishing attempts, especially given the increasing frequency and sophistication of malicious emails and internet of things (IoT) platforms. Attackers constantly hone their strategies by utilizing technology breakthroughs, highlighting the vital role those intelligent systems play in identifying and neutralizing new threats [1], [2]. This investigation explores the crucial do-main of phishing uniform resource locators (URLs), highlighting the necessity of utilizing machine-learning techniques to counter these dynamic threats [3]. Machine learning is showing promise as a powerful tool to improve the resilience of cybersecurity systems [4].

Malicious attackers use deceptive techniques in the context of phishing URLs to fool people into disclosing personal information or clicking on dangerous links. The increased susceptibility of linked devices to exploitation is a result of the growing attack surface that IoT platforms present. Innovative solutions are essential to differentiate between legitimate URLs and bad intent as phishing techniques, particularly HTML URL attacks, become more complex [5], [6]. With an emphasis on classification algorithms intended to identify harmful URLs, the research attempts to shed light on ideas and strategies to address the problems presented by imbalanced datasets [7], [8].

The Indian computer emergency response team (CERT-In), which is the official organization tasked with handling cybersecurity threats in the nation, handled 1,391,457 occurrences in 2022. With 875,892 occurrences, this represented the largest share of cases using vulnerable service mitigation. CERT-In responded to 19,793 reports of damage resulting from website defacements, which are major cybersecurity events in which attackers compromise and modify website content. It's surprising to note that 3,582 of these incidents affected websites with the.com domain, while 15,702 of them targeted websites with the.in domain. Phishing attacks increased significantly between 2021 and 2022, from 523 to 1,714 cases. This growth revealed a worrying trend in threat actor tactics targeted at civilians. Furthermore, from 728,276 in 2021 to 728,276 in 2022, a 20% increase in targeted vulnerable services were recorded.

High accuracy and quick response time can be achieved using techniques like deep neural network (DNN) [9] and variational autoencoders (VAE) [10], but they may overfit for specific dataset which may not capture possible variations of phishing attacks. CyberLen can combine factorization machine (FM) [2] and temporal convolutional networks (TCN) [11], however its interpretation of hidden correlations among lexical characteristics of the URLs is opaque. While hybrid models, like convolutional neural network (CNN)-long short-term memory (LSTM) [12], perform better in some applications, they need a lot of computing time. This study examined the limitations of existing methodologies and suggests a multi-algorithm approach to determine the most effective algorithm for phishing URL detection.

There are five sections in this paper. The introduction portion is shown in the first section. The literature review on the previous research is shown in the second section. Methodology, the third section, describes our techniques and methods used. Our results and findings are presented in results and discussion, the fourth section. The conclusion section describes about the summary and the future work.

2. LITERATURE REVIEW

A novel phishing detection system utilising DNN and VAE was presented by Prabakaran *et al.* [10]. The model overcame the drawbacks of blacklist-based techniques by achieving a quick reaction time of 1.9 seconds and a high accuracy of 97.45% on datasets including ISCX-URL-2016 and Kaggle. A new approach has been developed that uses a specialized neural network called a VAE to analyze raw URLs and automatically extract critical features. The model's noteworthy accuracy of 97.85% and outlining intentions to use generative modelling techniques in the future to lower the false-positive rate.

Phishing URL detection framework based on similarity index and incremental learning (PhiUSIIL), a groundbreaking system for phishing URL detection with incremental learning and a similarity index, was introduced by Prasad and Chandra [13]. It describes methods such as bit squatting, combo squatting, Punycode, homophone, homograph, and zero-width characters that are used in phishing attacks to deceive people visually. Various security profiles meet the requirements of users and organizations. In order to create a PhiUSIIL phishing URL dataset (134,850 authentic, 100,945 phishing URLs), the authors extracted characteristics. Experiments showed very high accuracy: 99.24% when the model was trained gradually and 99.79% when the model was first pre-trained. Regular knowledge updates guarantee flexibility in the face of new threats, highlighting the effectiveness of PhiUSIIL in dynamic cybersecurity.

CyberLen is a deep learning system that was introduced by Liang *et al.* [14]. This model uses an FM to find hidden correlations between lexical characteristics and a TCN to capture distant associations in harmful URLs. It overcomes the drawbacks of previous methods and reduces ambiguity by using position embedding for token vectorization. It uses a self-paced wide and deep learning approach to effectively integrate many aspects. When tested on an extensive URL data set, this has improved F1 scores and rate of convergence.

Nowroozi *et al.* [15] created a complex system for recognizing malicious advertisement URLs in order to mitigate cyber hazards. Compared to conventional ML strategies, we enhanced the model's resistance to obfuscation by using a new combination of lexical and web-scrapped features, spanning six classes. The framework achieved a low false negative rate of 0.0037 and a 99.63% detection accuracy by using machine learning techniques like random forest, gradient boost, XGBoost, and AdaBoost. The framework examined supervised learning using the K-Means algorithm for visual analysis and assessed how vulnerable decision tree-based models were to adversarial attacks like the Zeroth order optimisation adversarial attack.

Ogbuagu *et al.* [16] proposed a hybrid CNN-LSTM model to combat website URL spoofing in phishing attacks. Traditional methods, like blacklists and rule-based filters, prove inadequate against the increasing sophistication of phishing websites. They strategically combined CNN and LSTM models to overcome CNN's challenges in memorizing contextual relationships within URL text. Superior performance was established through evaluation utilising the UCL and PhishTank datasets, with accuracies of 98.9% and 96.8%, respectively, outperforming standalone CNN and LSTM. The hybrid model showcases efficacy in capturing nuanced features for improved detection accuracy, emphasizing its potential for advanced spoofing website URL detection.

DeepBF, a novel method for improved malicious URL detection that combines deep learning and a 2-dimensional bloom filter, was introduced by Patgiri *et al.* [17]. It is especially pertinent in light of the development of edge computing. Using a 2D structure and a carefully chosen non-cryptography string hash function. A biassed version of the hash function was developed for better performance, outperforming rival filters and a number of text hash functions that are not related to cryptography. The proposed method also includes an evolutionary CNN to detect counterfeit URLs to enhance detection capabilities. According to experimental data, the deepBF model is capable of effectively screening dangerous URLs on a wide range of devices, and Bloom Filter does not require cryptographic hash functions.

Karim *et al.* [18] deployed machine learning defences to counter the ubiquitous threat posed by phishing attempts. Based on a dataset of phishing URLs containing more than 11,000 website attributes, a variety of machine learning models, such as naive Bayes, decision trees, random forests, K-neighbours' classifiers, gradient boosting classifiers, support vector classifiers, and a new hybrid LSD model (LR+SVC+DT) are applied. Phishing attack prevention is demonstrated to be more accurate and efficient by the LSD model, which integrates both hard and soft voting. Evaluation measures that support the approach's efficacy include precision, accuracy, recall, F1-score, and specificity. It emphasizes internet privacy problems and emphasizes the critical role that machine learning plays in combating increasing cyber threats.

To overcome the difficulties associated with phishing identification, Rani *et al.* [19]. prioritize URL-based features and employ machine learning. To lessen the computing needs of external feature analysis and content, they develop TreeSHAP and information gain for feature selection. The seven-step procedure includes preprocessing, dataset partitioning, feature selection, cross-validation, model validation, and performance assessment. Remarkably, features delineated by TreeSHAP improve detection accuracy greatly; on the first dataset (15 features), XGBoost achieved 98.59%, and on the second dataset (20 features), random forest achieved 90.21%. Table 1 offers a comprehensive overview of algorithms utilized for detecting phishing URLs, drawing from previous contributions in the field. The table presents key details such as the algorithm employed, primary objectives of each study, and notable characteristics distinguishing each approach.

Table 1. Summary of literature survey

Author	Algorithm	Accuracy (%)	Data size	Application
Prabakaran *et al.* [10]	DNN, VAE	97.85	1.5 lakh URLs	URL
Prasad and Chandra [13]	Increamental learning	99.79	235,795 URLs	URL
Liang *et al.* [14]	TCN, FM	99.27	1,299,110 URL	URL
Nowroozi *et al.* [15]	RF, GB, XGB, and AdaBoost	99.63	3,980,870 URLs	URLs
Ogbuagu *et al.* [16]	CNN-LSTM	98.9	69,700 URLs	URLs
Patgiri *et al.* [17]	Evolutionary CNN	99.6	36,707 URLs	URLs
Karim *et al.* [18]	LR, SVC, DT	95.75	11,054 URLs	URLs
Rani *et al.* [19]	Naïve Bayes, random forest, and XGBoost	90.49	114,250 URLs	URLs
Proposed approach	Random forest, Adaboost, Catboost, and MLP	97.1	1,1053 URLs	URLs

3. METHODOLOGY

The Figure 1 depicts the architectural diagram of the proposed model. An extensive dataset comprising 11,053 records with 31 features is utilized. The dataset undergoes exploratory data analysis (EDA) and data cleaning procedures, where duplicate rows and erroneous values are addressed. Following data cleaning, the data preprocessing stage involves feature scaling and normalization. The dataset is then split into training and testing sets in a 70:30 ratio for model development and training. Random forest, Catboost, AdaBoost, and multi-layer perceptron (MLP) algorithms are employed for model training. Subsequently, the trained model is utilized to classify URLs as legitimate or phishing. The performance of the trained model is evaluated using metrics such as accuracy, precision, F1-score, and recall. The proposed methodology consists of 4 modules known as data cleaning, data pre-processing, model development and model evaluation.

3.1. Dataset description

The dataset, sourced from Kaggle, includes 11,053 instances of phishing site URL identification as shown in Figure 2. Of these, 6,155 records are legitimate and 4,898 are phishing. Each instance is defined by 32 features, including 'LongURL', 'ShortURL', 'Symbol@', 'Redirecting//', 'AgeofDomain', and 'SubDomains', which provide insights into URL characteristics indicative of phishing attempts.

3.2. Algorithm for proposed approach

This alogithm begins with loading the dataset followed by data cleaning. Data cleaning includes handling duplicate rows and error values in the dataset. Data processing is then applied to the cleaned dataset by performing feature scaling and normalization techniques. The final dataset splits into training and testing sets, and multiple models are trained. The model with highest accuracy is selected as the final classifier for accurate predictions.

Algorithm: Proposed approach

```
Input: Dataset (features, target)
Step 1: Load the Dataset
Step 2: Data Cleaning:
Step 2.1: Handling duplicate rows
g=group(duplicate_rows)
For each g:
    Keep one instance and mark the others as duplicate.
Step 2.2: Correcting erroneous values:
For each f ∈ features:
    R=Range(f)
for each r ∈ row:
    for v ∈ features:
    if v not in R:
    set v=Min_value(R)
Step3: Data Preprocessing:
    Step 3.1: Feature Scaling:
        For X ∈ feature:
        $X_{min}$=Min_value(f)
        $X_{max}$=Max_value(f)

    $X_{scaled} = \frac{X_{max}-X_{min}}{X-X_{min}}$

    Step 3.2: Feature Normalization
For X ∈ feature:

    $X_{norm} = \frac{X-\mu}{\sigma}$
Step 4: Model Development:
    Tr,Ts=Split(dataset)
    Rf_model=RandomForest(Tr)
    Ab_model=AdaBoost(Tr)
    Cb_model=CatBoost(Tr)
    Mlp_model=MultilayerPerceptron(Tr)
    Classifier_model=Max_accuracy(Rf_model,Ab_model,Cb_model,Mlp_model)
    prediction=Predict(url)
```

3.2.1. Random forest

For phishing URL identification, random forest [20] is an ensemble learning technique. Using distinct feature sets and subsets of the dataset, builds numerous decision trees, each concentrating on a different set of attributes. This model makes predictions by integrating the outputs of various trees and efficiently differentiating between phishing and legitimate URLs. To improve decision-making at every node and increase the precision of phishing attempt detection, the model makes use of measures such as Gini impurity.

```
Input:
Dataset D= (features, label)
Number of trees (N)
Number of features (NF)

Algorithm: Random Forest
For i = 1 to N:
     Create a random bootstrapped dataset
     $D: D_b = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n , y_n)\}$
     Randomly select a subset of features for the tree $F_s = \{f_1, f_2, \ldots, f_{NF}\}$
     Build a decision tree using the bootstrapped dataset $D_b$ and the selected features $F_s$
At each node, use Gini impurity for classification:
 Gini Impurity $= 1 - \sum_{j=1}^{C} p_j^2$ where C = number of classes
 Final Prediction $= argmax_c(\sum_{k=1}^{N} Pred_{k=c})$
```

3.2.2. AdaBoost

AdaBoost [21] creates a powerful model by merging many weak classifiers to detect phishing URLs. To improve detection accuracy, this model repeatedly modifies the weights of incorrectly identified URLs to concentrate on more difficult situations. The final forecast is the weighted aggregate of all learners, with the effect of each weak learner determined by its performance.

```
Algorithm: AdaBoost
Initialize weights $w_i$=1/N
For t=I to T:
    Train a week learner ht(x) with weights wi.
    Calculate the weighted error of each weak learner as $\varepsilon_t = \sum_{i=1}^{N} w_i \cdot 1(hf_t(x_i)! = y_i)$
    Compute the weight of the weak learner as $\alpha_t = \frac{1}{2} \ln \left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$
    Update the weights as $w_i = w_i \cdot e^{(-\alpha_t . y_i . hf_t(xi))}$
    Normalize the weights: $w_i = \frac{w_i}{\sum_{i=1}^{N} w_i}$
    Combine the week classifiers into strong classifier: $H(x) = sign(\sum_{i=1}^{N} \alpha_t (hf_t(x)))$
Final prediction: $sign(\sum_{i=1}^{T} \alpha t \cdot hf_t(x_{new}))$
```

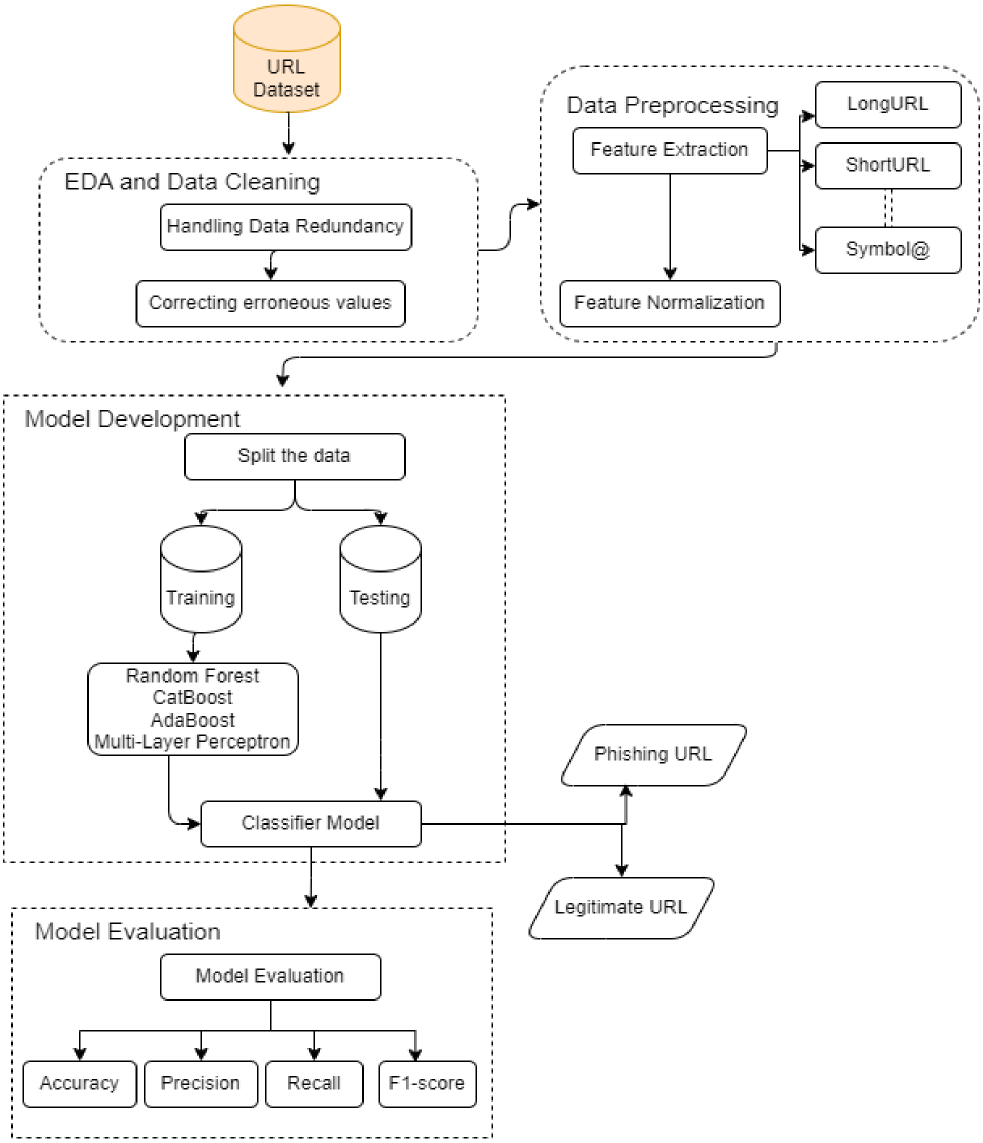

Figure 1. Proposed architecture

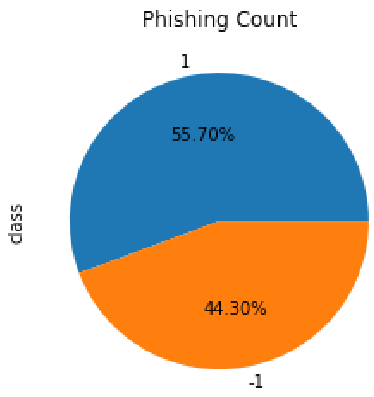

Figure 2. Dataset description

3.2.3. CatBoost

CatBoost [22] efficiently handles categorical information and repeatedly constructs decision trees to enhance phishing URL detection. In order to improve accuracy, it concentrates on challenging scenarios while updating predictions using calculated negative gradients. Boosted scores are converted into probabilities in the final forecast, which shows the possibility that a URL is phishing.

```
Algorithm: CatBoost
Objective function: $Logloss(L) = -\frac{1}{N}\sum_{i=1}^{N}[y_i log(p_i) + (1 - y_i)log(1 - p_i)]$
Initialize predictions as $F_0(x_i) = \log\left(\frac{p}{1-p}\right)$
For t= 1 to t:
    Compute Negative gradient: $g_{it} = \frac{\partial L(y_i, p_i)}{\partial F_{t-1}(x_i)}$
    Built a decision tree using $g_{it}$
    Output for the k-th leaf in the t-th tree: $\gamma_{kt} = -\frac{\sum_{i \in leaf\ k} g_{it}}{\sum_{i \in leaf\ k}|g_{it}| + \lambda}$
    Update the predictions: $F_t(x_i) = F_{t-1}(x_i) + \vartheta . \gamma_{kt}$
Final prediction: $p_i = \frac{1}{1+e^{-F_T(x_i)}}$
```

3.2.4. Multi-layer perceptron

For the purpose of phishing URL detection, a MLP [23] generates a likelihood score by processing URL information over many layers. In order to reduce binary cross-entropy loss, iteratively updating weights and biases via backpropagation is employed. By using this method, the model becomes more accurate at detecting phishing URLs.

```
Algorithm: Multi layer perceptron
    Initialize the weights(w) and biases(b) in the network.
    for each epoch:
        Perform Forward Propagation as follows:
                for each training sample $x_i$:
                        Set the input layer $a^{(0)}$as the feature vector xi
                        for each hidden layer l:
                                Weighed sum: $z^{(l)} = W^{(l)} * a^{(l-1)} + b^{(l)}$
                                Activation function: $a^{(l)} = activation(z^{(l)})$
                for the output layer (L):
                        Final weighted sum: $z^{(L)} = W^{(L)} * a^{(L-1)} + b^{(L)}$
                        Output activation function: $\hat{y} = sigmoid(z^{(L)})$
        Compute binary cross-entropy loss as $Loss = -\frac{1}{N}\sum_{i=1}^{N}[y_i log(\widehat{y_\imath}) + (1 - y_i)log(1 - \hat{y}_i)]$
        Perform backward Propagation as follows:
                Output layer: $\delta^{(L)} = \hat{y}_i - y$
                Hidden layer: $\delta^{(l)} = (W^{(l+1)})^T * \delta^{(l+1)} \odot \sigma(z^{(l)}).(1 - \sigma(z^{(l)}))$
```

Update gradients: $\frac{\partial Loss}{\partial W^{(l)}} = \delta^{(l)} * (a^{(l-1)})^T$ and $\frac{\partial Loss}{\partial b^{(l)}} = \delta^{(l)}$

Update weights and biases: $W^{(l)}{}_{new} = W^{(l)}{}_{old} - \alpha * \frac{\partial Loss}{\partial W^{(l)}}$ and $b^{(l)}{}_{new} = b^{(l)}{}_{old} - \frac{\partial Loss}{\partial b^{(l)}}$

4. RESULTS AND DISCUSSION

Figure 3 depicts a heat map illustrating variable relationships via correlation analysis. In this heat map, both the x-axis and y-axis correspond to the features of the dataset. Color intensity denotes the strength and direction of correlations, aiding in identifying patterns. The visualization offers insights into dataset interdependencies, facilitating analysis, and decision-making.

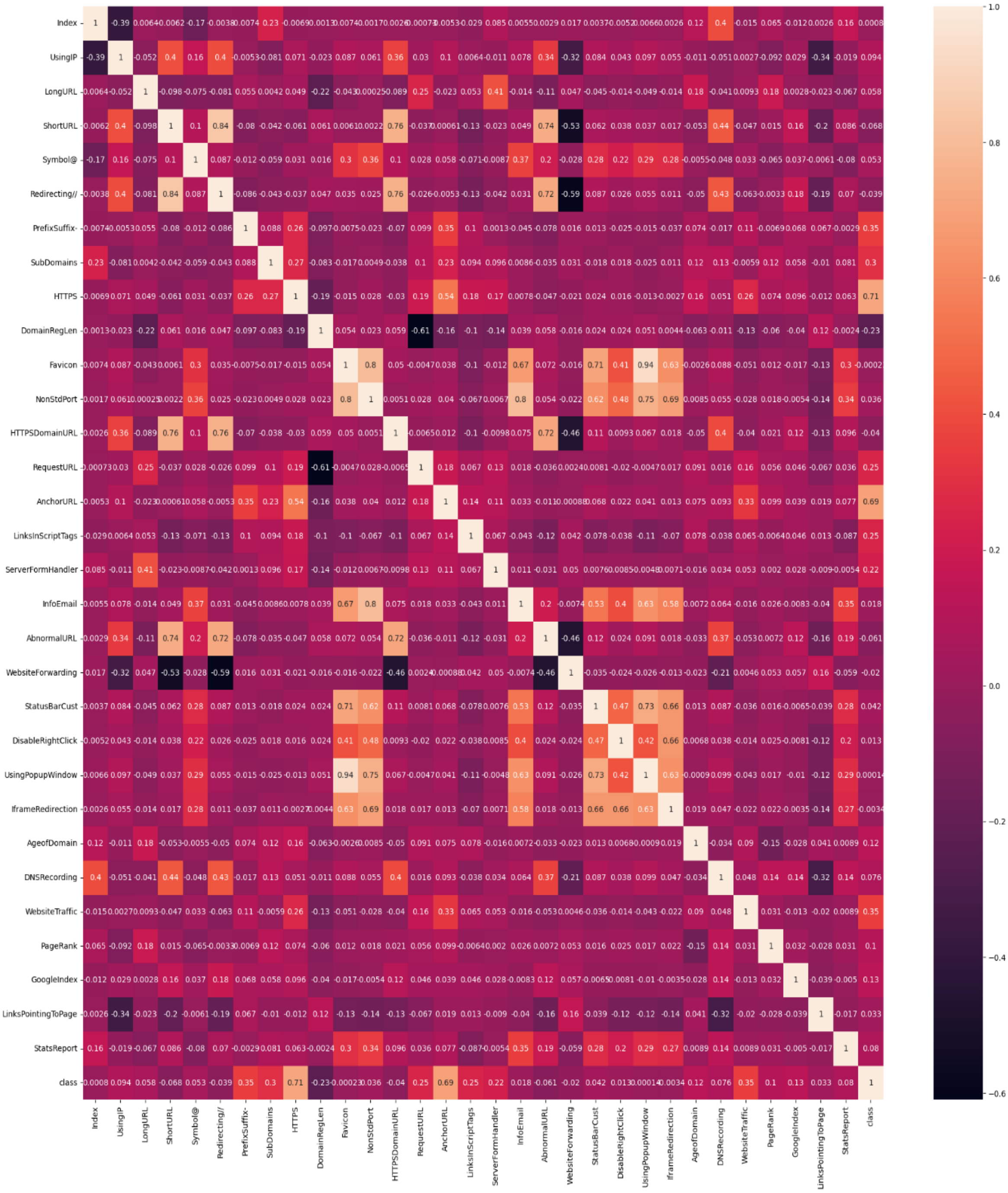

Figure 3. Heat map of the input variables

4.1. Performance analysis

The Table 2 presents the comparision in performance metrics of four distinct machine learning models focusing on accuracy, F1 score, recall, and precision. The four ML models used are random forest, CatBoost classifier, MLP [24], and AdaBoost. The MLP and CatBoost classifier shows the best performance followed by random forest. While AdaBoost classifier has lower accuracy, still it performs well in recall and F1 score.

Table 2. Evaluation metrics

ML Model	Accuracy	f1 score	Recall	Precision
Random forest	0.967	0.970	0.992	0.991
CatBoost classifier	0.972	0.975	0.994	0.989
AdaBoost	0.938	0.945	0.957	0.935
Multilayer perceptron	0.971	0.974	0.992	0.985

4.1.1. Confusion matrix

Confusion matrix shows the performance of the model by comparing the actual and predicted class labels. Figures 4 and 5 depict the confusion matrices for the random forest and AdaBoost classifiers, demonstrating their effectiveness in accurately predicting [25] both positive and negative instances. The CatBoost and MLP confusion matrix visually summarizes the neural network's classification performance, showcasing accurate predictions [26] in the diagonal elements as shown in Figures 6 and 7.

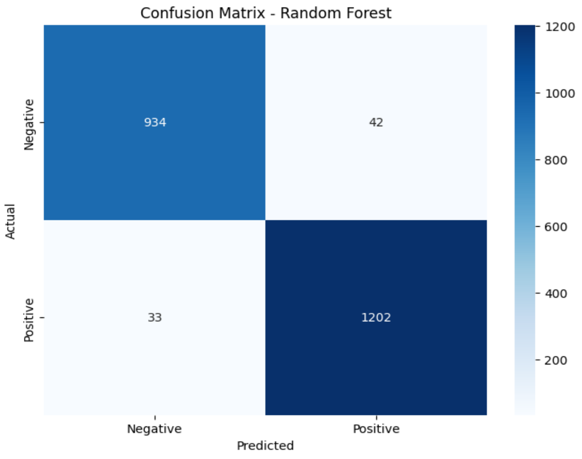

Figure 4. Random forest

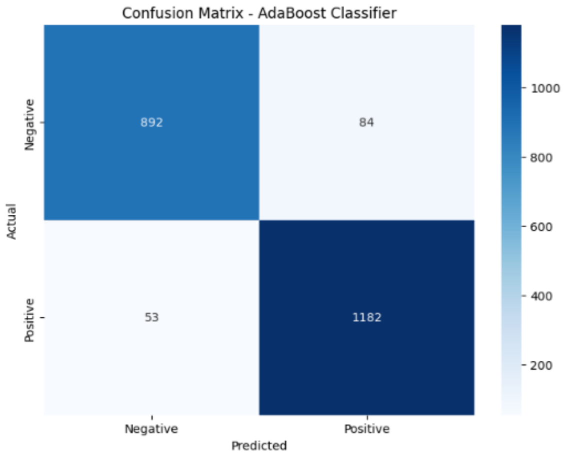

Figure 5. AdaBoost

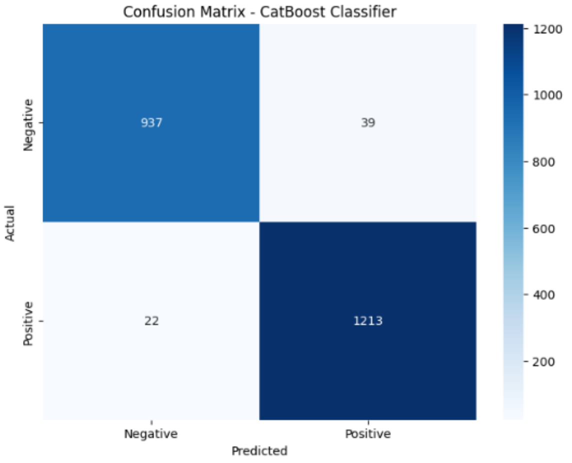

Figure 6. CatBoost

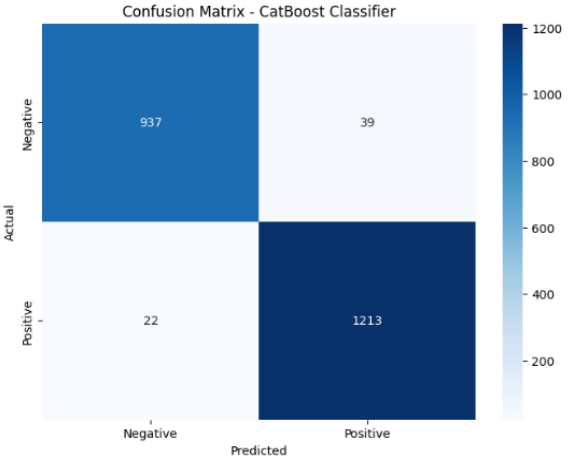

Figure 7. Multi-layer perceptron

5. CONCLUSION

This study employed feature extraction and feature normalization approaches to transform input URLs into meaningful numerical values, which led to the development of an efficient system for identifying phishing URLs. Our findings demonstrate that combining these methods with a MLP greatly increases the accuracy of identifying legitimate from phishing URLs. The suggested approach is dependable and strong, indicating that performance may be improved by more optimization and integration with other machine learning techniques. Further comprehensive research is necessary to validate its efficacy against various phishing attack vectors and URL modifications.

REFERENCES

[1] A. Maci, A. Santorsola, A. Coscia, and A. Iannacone, "Unbalanced web phishing classification through deep reinforcement learning," *Computers*, vol. 12, no. 6, 2023, doi: 10.3390/computers12060118.

[2] S. R. Abdul Samad et al., "Analysis of the performance impact of fine-tuned machine learning model for phishing URL detection," *Electronics*, vol. 12, no. 7, 2023, doi: 10.3390/electronics12071642.

[3] A. Alanazi and A. Gumaei, "A decision-fusion-based ensemble approach for malicious websites detection," *Applied Sciences*, vol. 13, no. 18, 2023, doi: 10.3390/app131810260.

[4] S. Mohanty and A. A. Acharya, "MFBFST: Building a stable ensemble learning model using multivariate filter-based feature selection technique for detection of suspicious URL," *Procedia Computer Science*, vol. 218, pp. 1668–1681, 2022, doi: 10.1016/j.procs.2023.01.145.

[5] R. Roopalakshmi, A. Shukla, J. Karthikeyan, and K. Banerjee, "Robust framework for malevolent URL detection using hybrid supervised learning," *Procedia Computer Science*, vol. 230, pp. 241–247, 2023, doi: 10.1016/j.procs.2023.12.079.

[6] S. S. Shin, S. G. Ji, and S. S. Hong, "A heterogeneous machine learning ensemble framework for malicious webpage detection," *Applied Sciences*, vol. 12, no. 23, 2022, doi: 10.3390/app122312070.

[7] M. Aljabri *et al.*, "Detecting malicious URLs using machine learning techniques: review and research directions," *IEEE Access*, vol. 10, pp. 121395–121417, 2022, doi: 10.1109/ACCESS.2022.3222307.

[8] C. R. Vyawhare, R. Y. Totare, P. S. Sonawane, and P. B. Deshmukh, "Machine learning system for malicious website detection: a literature review," *International Journal for Research in Applied Science and Engineering Technology*, vol. 10, no. 5, pp. 56–61, 2022, doi: 10.22214/ijraset.2022.42050.

[9] A. Ozcan, C. Catal, E. Donmez, and B. Senturk, "A hybrid DNN–LSTM model for detecting phishing URLs," *Neural Computing and Applications*, vol. 35, no. 7, pp. 4957–4973, 2023, doi: 10.1007/s00521-021-06401-z.

[10] M. K. Prabakaran, P. M. Sundaram, and A. D. Chandrasekar, "An enhanced deep learning-based phishing detection mechanism to effectively identify malicious URLs using variational autoencoders," *IET Information Security*, vol. 17, no. 3, pp. 423–440, 2023, doi: 10.1049/ise2.12106.

[11] S. Gopali, A. S. Namin, F. Abri, and K. S. Jones, "The performance of sequential deep learning models in detecting phishing websites using contextual features of URLs," *Proceedings of the ACM Symposium on Applied Computing*, pp. 1064–1066, 2024, doi: 10.1145/3605098.3636164.

[12] Z. Alshingiti, R. Alaqel, J. Al-Muhtadi, Q. E. U. Haq, K. Saleem, and M. H. Faheem, "A deep learning-based phishing detection system using CNN, LSTM, and LSTM-CNN," *Electronics*, vol. 12, no. 1, 2023, doi: 10.3390/electronics12010232.

[13] A. Prasad and S. Chandra, "PhiUSIIL: A diverse security profile empowered phishing URL detection framework based on similarity index and incremental learning," *Computers and Security*, vol. 136, 2024, doi: 10.1016/j.cose.2023.103545.

[14] Y. Liang, Q. Wang, K. Xiong, X. Zheng, Z. Yu, and D. Zeng, "Robust detection of malicious URLs with self-paced wide & deep learning," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 2, pp. 717–730, 2022, doi: 10.1109/TDSC.2021.3121388.

[15] E. Nowroozi, Abhishek, M. Mohammadi, and M. Conti, "An adversarial attack analysis on malicious advertisement URL detection framework," *IEEE Transactions on Network and Service Management*, vol. 20, no. 2, pp. 1332–1344, 2023, doi: 10.1109/TNSM.2022.3225217.

[16] B. C. U. -Ogbuagu, O. N. Akande, and E. Ogbuju, "A hybrid deep learning technique for spoofing website URL detection in real-time applications," *Journal of Electrical Systems and Information Technology*, vol. 11, no. 1, 2024, doi: 10.1186/s43067-023-00128-8.

[17] R. Patgiri, A. Biswas, and S. Nayak, "deepBF: Malicious URL detection using learned bloom filter and evolutionary deep learning," *Computer Communications*, vol. 200, pp. 30–41, 2023, doi: 10.1016/j.comcom.2022.12.027.

[18] A. Karim, M. Shahroz, K. Mustofa, S. B. Belhaouari, and S. R. K. Joga, "Phishing detection system through hybrid machine learning based on URL," *IEEE Access*, vol. 11, pp. 36805–36822, 2023, doi: 10.1109/ACCESS.2023.3252366.

[19] L. M. Rani, C. F. M. Foozy, and S. N. B. Mustafa, "Feature selection to enhance phishing website detection based on URL using machine learning techniques," *Journal of Soft Computing and Data Mining*, vol. 4, no. 1, pp. 30–41, 2023, doi: 10.30880/jscdm.2023.04.01.003.

[20] S. Sindhu, S. P. Patil, A. Sreevalsan, F. Rahman, and A. N. Saritha, "Phishing detection using random forest, SVM and neural network with backpropagation," *Proceedings of the International Conference on Smart Technologies in Computing, Electrical and Electronics, ICSTCEE 2020*, pp. 391–394, 2020, doi: 10.1109/ICSTCEE49637.2020.9277256.

[21] F. Nthurima, A. Mutua, and W. Stephen Titus, "Detecting phishing emails using random forest and AdBoost classifier model," *Open Journal for Information Technology*, vol. 6, no. 2, pp. 123–136, 2023, doi: 10.32591/coas.ojit.0602.03123n.

[22] K. Sadaf, "Phishing website detection using XGBoost and Catboost classifiers," *International Conference on Smart Computing and Application, ICSCA 2023*, 2023, doi: 10.1109/ICSCA57840.2023.10087829.

[23] A. U. Z. Asif, H. Shirazi, and I. Ray, "Machine learning-based phishing detection using URL features: a comprehensive review," *Stabilization, Safety, and Security of Distributed Systems (SSS 2023)*, pp. 481–497, 2023, doi: 10.1007/978-3-031-44274-2_36.

[24] S. Remya, M. J. Pillai, K. K. Nair, S. R. Subbareddy, and Y. Y. Cho, "An effective detection approach for phishing URL using ResMLP," in *IEEE Access*, vol. 12, pp. 79367-79382, 2024, doi: 10.1109/ACCESS.2024.3409049.

[25] A. S. Rafsanjani, N. B. Kamaruddin, M. Behjati, S. Aslam, A. Sarfaraz, and A. Amphawan, "Enhancing malicious URL detection: A novel framework leveraging priority coefficient and feature evaluation," in *IEEE Access*, vol. 12, pp. 85001-85026, 2024, doi: 10.1109/ACCESS.2024.3412331.

[26] M. Sameen, K. Han, and S. O. Hwang, "PhishHaven—an efficient real-time AI phishing URLs detection system," in *IEEE Access*, vol. 8, pp. 83425-83443, 2020, doi: 10.1109/ACCESS.2020.2991403.

BIOGRAPHIES OF AUTHORS

Devi Sree Guddanti is a final-year B.Tech. student, specializing in Computer Science and Engineering at V. R. Siddhartha Engineering College, Vijayawada, India. she is passionate about artificial intelligence and machine learning. She can be contacted at email: devisree1293@gmail.com.

Dr. Rupa Chiramdasu is currently working as a Professor in the Department of Computer Science and Engineering, V. R. Siddhartha Engineering College, Vijayawada, India. She received Post Doctoral Fellowship in Computer Science and Engineering in 2021 from University of Lincoln, Malaysia. She has 19 years of teaching experience. Her research interests lie in areas such as information security, computational intelligence based security, and blockchain technology. She published 136 national and international articles and got a patent granted with 4 published patents. She can be contacted at email: rupamtech@gmail.com.

Mahalakshmi Gayathri Uppuluri is a final-year B.Tech. student, specializing in Computer Science and Engineering at V. R. Siddhartha Engineering College, Vijayawada, India. She is passionate about artificial intelligence and machine learning. She can be contacted at email: umlgayathri77@gmail.com.