Journal of
Cyber Security

Tech Science Press

DOI: 10.32604/jcs.2024.048332

ARTICLE

Securing Web by Predicting Malicious URLs

Imran Khan and Meenakshi Megavarnam[*]

School of Computer Science, University of Hertfordshire, Hatfield, AL10 9AB, UK

*Corresponding Author: Meenakshi Megavarnam. Email: meenaaksharadn@gmail.com

Received: 05 December 2023 Accepted: 08 November 2024 Published: 06 December 2024

ABSTRACT

A URL (Uniform Resource Locator) is used to locate a digital resource. With this URL, an attacker can perform a variety of attacks, which can lead to serious consequences for both individuals and organizations. Therefore, attackers create malicious URLs to gain access to an organization's systems or sensitive information. It is crucial to secure individuals and organizations against these malicious URLs. A combination of machine learning and deep learning was used to predict malicious URLs. This research contributes significantly to the field of cybersecurity by proposing a model that seamlessly integrates the accuracy of machine learning with the swiftness of deep learning. The strategic fusion of Random Forest (RF) and Multilayer Perceptron (MLP) with an accuracy of 81% represents a noteworthy advancement, offering a balanced solution for robust cybersecurity. This study found that by combining RF and MLP, an efficient model was developed with an accuracy of 81% and a training time of 33.78 s.

KEYWORDS

Malicious URLs; prediction; machine learning; deep learning; random forest; multilayer perceptron; securing web

1 Introduction

One cannot avoid the use of the internet in today's world. Therefore, web security has become critical in securing individuals and organizations from cyber threats. By implementing stringent and proactive web security measures, companies can protect their online environment and provide a safe place for their consumers [1]. When workers have access to dangerous files and websites, a firewall, intrusion prevention system, URL filtration, and access restrictions can be implemented to reduce the company's risk [2]. There are various kinds of online assaults like cross-site scripting, SQL injection, phishing, denial of service, and many more. The main goal of cyber criminals, who pose as legitimate website, is to access sensitive data and systems from a company or individual for financial gain.

Malicious URLs are one the important factors through which many cyber-attacks occur. Uniform resource locator (URL) is the permanent address for the resources. These resources can be of any kind, such as files, audio, and images. Protocol controls the information transmission in a network. The resource ID is placed after the URL resource type [3]. The URL is used to find information from a specific place on the World Wide Web. The primary objective of URL filtering is to prevent online assaults, therefore strengthening cybersecurity for organizations and individuals [4].

The motivation for this research comes from the ever-evolving landscape of cyber threats, particularly those involving malicious URLs. URLs pose a significant challenge, threatening individuals and organizations with data breaches, financial losses, and reputation damage. This research makes a substantial contribution to the field of cybersecurity by focusing on preventing malicious URLs. The proposed fusion of RF (Random Forest) and MLP (Multilayer Perceptron) represents a novel advancement, offering a balanced solution for predicting malicious URLs. This kind of attack can be prevented by implementing a model for predicting the malicious URLs.

2 Literature Review

The research paper titled "Detecting Malicious URLs Using Binary Classification Through Adaboost Algorithm" [5] uses machine learning to create a comprehensive prototype to predict malicious URLs. This research focuses on exploring the perfect formulation for finding malicious URLs using machine learning. It also introduces an approach to leverage the Adaboost algorithm. Adaboost was selected due to its flexibility, as it can be combined with other machine learning algorithms. Fig. 1 is the result of this research paper that highlights the number of malware occurrences classified into different categories.

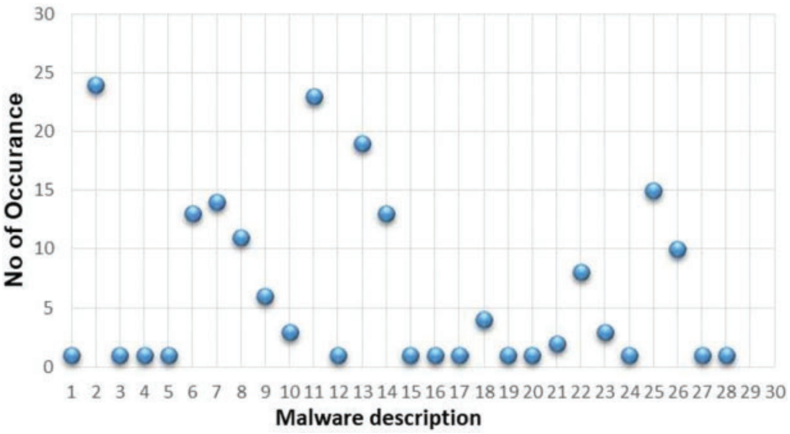
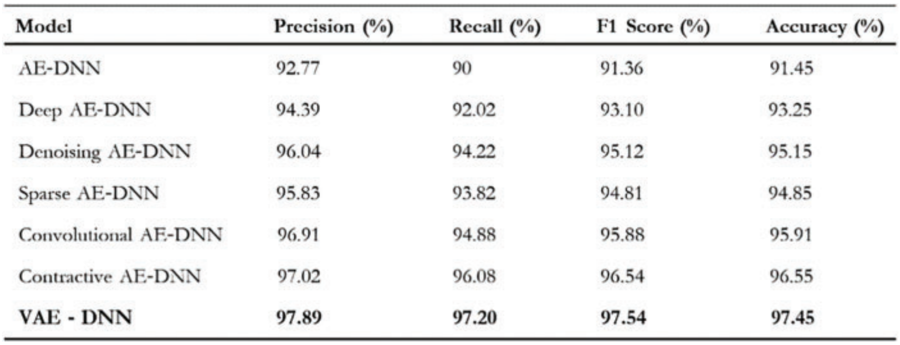


Figure 1: Results of Khan, F. (Reprinted from Reference [5])

The research paper, "An Enhanced Deep Learning-Based Phishing Detection Mechanism to Effectively Identify Malicious URLs Using Variational Autoencoders" [6], focuses on introducing an advanced deep learning-based model for predicting phishing. This enhances the overall capability of the model to predict malicious URLs. This paper combines the strengths of Variational Autoencoders (VAE) and Deep Neural Networks (DNN) to capture the intrinsic features of URLs, thereby enhancing the model's ability to identify phishing URLs. The dataset used in this research contained 100,000 URLs from two open sources: the ISCX-URL-2016 dataset and the Kaggle dataset. The proposed model achieved an accuracy of 97.45%, with a response time of 1.9 s, which is superior to all other evaluated models.

Fig. 2, which is the result of this research, clearly states the performance of all the models in terms of precision, recall, F1 score, and accuracy. Among all the models, the VAE-DNN model outperformed the others with an accuracy of 97.45%.

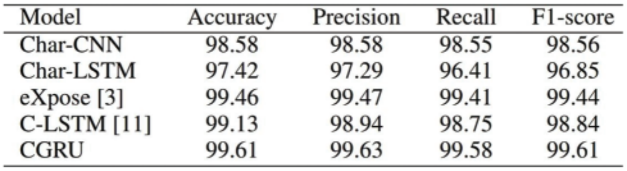


Model	Precision (%)	Recall (%)	F1 Score (%)	Accuracy (%)
AE-DNN	92.77	90	91.36	91.45
Deep AE-DNN	94.39	92.02	93.10	93.25
Denoising AE-DNN	96.04	94.22	95.12	95.15
Sparse AE-DNN	95.83	93.82	94.81	94.85
Convolutional AE-DNN	96.91	94.88	95.88	95.91
Contractive AE-DNN	97.02	96.08	96.54	96.55
VAE - DNN	**97.89**	**97.20**	**97.54**	**97.45**

Figure 2: Results of Prabakaran, M. K. (Reprinted from Reference [6])

The research paper by Aljabri et al. [7] focused on the literature review of existing papers related to finding malicious URLs using machine learning in both Arabic and non-Arabic content. It mainly focused on key findings, specifically the use of lexical features in a URL to predict malicious content. It was also found that there was a recurrent use of Support Vector Machine (SVM), Random Forest (RF), and Naïve Bayes (NB) in the reviewed papers. Additionally, the performance of the Convolutional Neural Network (CNN) and XGBoost models was exceptional, with an accuracy of 99.98%.

In the research paper, "Detecting Malicious URLs via a Keyword-Based Convolutional Gated-Recurrent-Unit Neural Network" [8], a Convolutional Gated Recurrent Unit (CGRU) neural network was introduced to predict malicious URLs. This model features character-based text classification. The traditional pooling layer is replaced with the Gated Recurrent Unit (GRU) to enhance the capturing of temporal features in the URL. Fig. 3 shows the comparison results of all the models on the test set. The comparison reveals that the CGRU model has the highest accuracy of 99.6%, outperforming all other models.

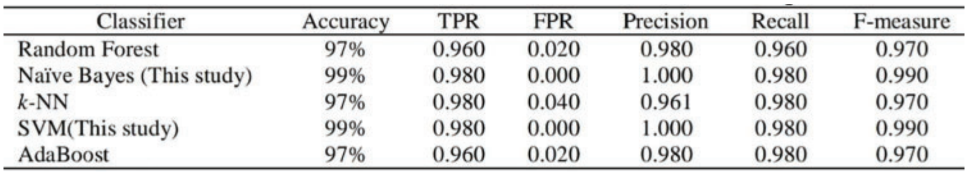


Model	Accuracy	Precision	Recall	F1-score
Char-CNN	98.58	98.58	98.55	98.56
Char-LSTM	97.42	97.29	96.41	96.85
eXpose [3]	99.46	99.47	99.41	99.44
C-LSTM [11]	99.13	98.94	98.75	98.84
CGRU	99.61	99.63	99.58	99.61

Figure 3: Results of Yang, W. (Reprinted from Reference [8])

The research paper, "A Malicious URLs Detection System Using Optimization and Machine Learning Classifiers" [9], aims to assess the efficiency of machine learning models in predicting malicious URLs. It uses a bio-inspired algorithm, Particle Swarm Optimization (PSO), which is a feature optimization method to select critical URL attributes for detecting malicious URLs. By combining machine learning and static analysis, this study improves prediction accuracy.

Fig. 4 demonstrates the detection performance of the five classifiers used in this research. The performance is evaluated in terms of accuracy, true positive rate (TPR), false positive rate (FPR), precision, recall, and F-measure. Fig. 4 indicates that the Naïve Bayes and SVM have the highest precision in predicting the malicious dataset.



Classifier	Accuracy	TPR	FPR	Precision	Recall	F-measure
Random Forest	97%	0.960	0.020	0.980	0.960	0.970
Naïve Bayes (This study)	99%	0.980	0.000	1.000	0.980	0.990
k-NN	97%	0.980	0.040	0.961	0.980	0.970
SVM(This study)	99%	0.980	0.000	1.000	0.980	0.990
AdaBoost	97%	0.960	0.020	0.980	0.980	0.970

Figure 4: Results of Lee, O. V. (Reprinted from Reference [9])

In the research papers listed in Table 1, the authors have used either machine learning or deep learning to detect malicious URLs. However, this research paper combines both machine learning and deep learning algorithms to leverage the strengths of both: the accuracy of machine learning and the swiftness of deep learning. Thus, this proposed method can be more efficient in detecting malicious URLs.

Table 1: Literature review comparison

Reference	Approach/model	Accuracy
[5]	AdaBoost algorithm	Higher than other ML algorithms
[6]	VAE and deep neural network	97.45%
[7]	CNN with XGBoost	99.98%
[8]	CGRU neural network	99.61%
[9]	Machine learning + Static analysis (NB, SVM)	99%

3 Aim

This research paper aims to implement an efficient model to predict malicious URLs in the categories of phishing, malware, and defacement. It will use different machine learning and deep learning algorithms to build a system for predicting malicious URLs. Furthermore, it will also combine machine learning and deep learning to determine if this hybrid approach is more efficient than individual algorithms.

4 Research Question

Which algorithm is best to find malicious URLs in the categories of Benign, Phishing, Malware, and defacement?

5 Proposed Method

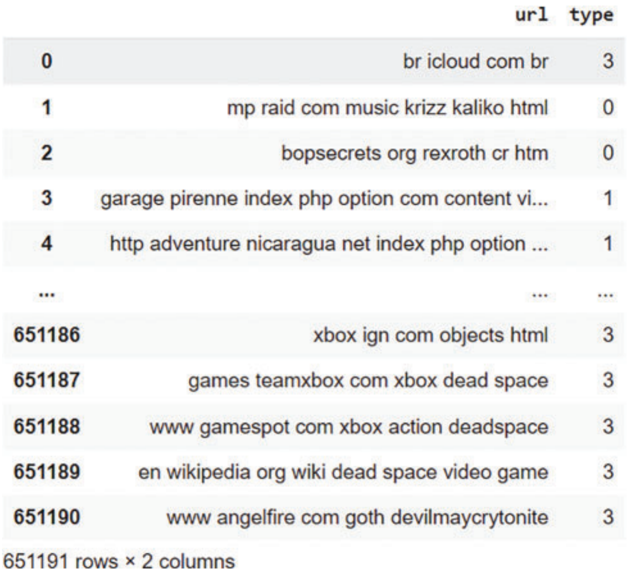
This research paper presents a combination of Random Forest (machine learning) and Multilayer Perceptron (deep learning) algorithms to predict malicious URLs in the categories of benign, phishing, malware, and defacement. This combination leverages the accuracy of machine learning and the swiftness of deep learning to create a more efficient model for detecting malicious URLs. This study emphasizes the benefits of the combined model.

6 Dataset

This study utilizes the dataset from Kaggle [10], which is open-source. The dataset comprises 651,191 URLs categorized as benign, malware, defacement, and phishing. Machine learning and deep learning models are trained using this dataset to detect malicious URLs, thereby enhancing web security. The dataset consists of two columns: URL and type.

7 Preprocessing

The dataset was preprocessed by removing all null values, duplicate values, and stop words before usage. Additionally, label encoding was applied to replace the “type” column with integers, as illustrated in Fig. 5.



	url	type
0	br icloud com br	3
1	mp raid com music krizz kaliko html	0
2	bopsecrets org rexroth cr htm	0
3	garage pirenne index php option com content vi...	1
4	http adventure nicaragua net index php option ...	1
...	...	...
651186	xbox ign com objects html	3
651187	games teamxbox com xbox dead space	3
651188	www gamespot com xbox action deadspace	3
651189	en wikipedia org wiki dead space video game	3
651190	www angelfire com goth devilmaycrytonite	3

651191 rows × 2 columns

Figure 5: Label encoded malicious URL dataset

The type column is encoded as follows:

0—benign

1—defacement

2—malware

3—phishing

Now, the preprocessed dataset is used in this research.

Visualizations of the URL types are also presented in Fig. 6.

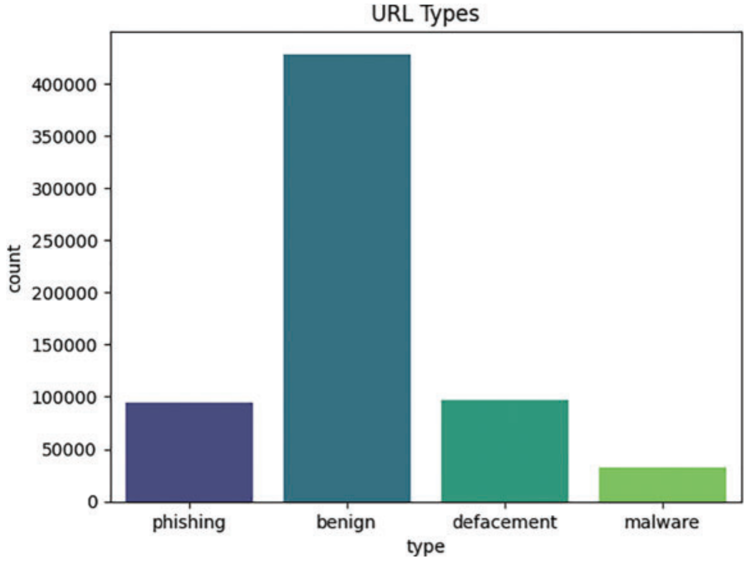


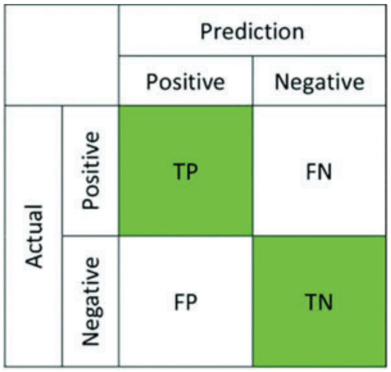
Figure 6: Visualization of URL types

8 Evaluation Metrics

This section focuses on evaluating different algorithms using key metrics such as accuracy, F1 score, precision, and recall.

8.1 Confusion Matrix

The confusion matrix provides clarity on true positives, true negatives, false positives, and false negatives generated by the models. It illustrates how effectively the model distinguishes between true and false malicious URLs [11]. Fig. 7 presents the confusion matrix, which visually represents the performance of binary classification



		Prediction	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

Figure 7: Confusion matrix for performance evaluation (Reprinted from Reference [12])

8.2 *Precision*

Precision indicates the algorithm's effectiveness in categorizing URLs into groups such as benign, phishing, malware, and defacement.

8.3 *Recall*

Recall demonstrates the model's ability to accurately predict actual malicious URLs, thereby minimizing false negatives. It plays a crucial role in providing comprehensive predictions [13].

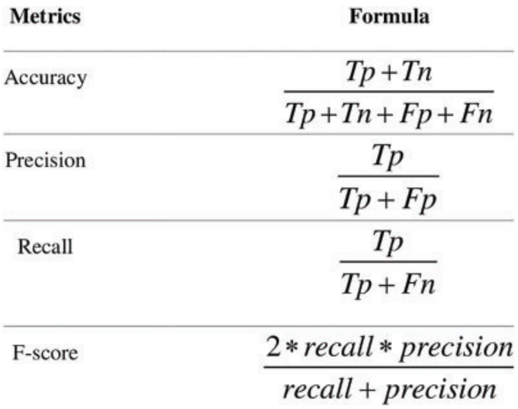
8.4 *F1 Score*

It provides an overall evaluation of the machine learning and deep learning models which includes precision and recall. This metric estimates the model's capability to balance between accuracy and completeness [14].

8.5 *Accuracy*

Accuracy measures the correctness of the model in predicting malicious URLs. It represents the ratio of correctly predicted malicious URLs to the total predicted URLs [15].

Fig. 8 displays the mathematical formulas for these performance metrics, which are used to compare the results of different models.



Metrics	Formula
Accuracy	$\frac{Tp+Tn}{Tp+Tn+Fp+Fn}$
Precision	$\frac{Tp}{Tp+Fp}$
Recall	$\frac{Tp}{Tp+Fn}$
F-score	$\frac{2*recall*precision}{recall+precision}$

Figure 8: Performance metrics (Reprinted from Reference [16])

8.6 *Training Time*

It indicates the time taken by the deep learning and machine learning models to learn from the data. This gives an insight into the practical and effective aspects of the model [17].

8.7 *Validation Time*

Validation time refers to the duration taken to evaluate the overall performance of the training model on the validation dataset.

8.8 Testing Time

Testing time refers to the duration taken by the model to predict malicious URLs from new data. This metric is crucial for evaluating the responsiveness of machine learning or deep learning models in real-time scenarios [18].

9 Results and Evaluation

This section discusses the results, evaluation, and comparison of different models.

9.1 Machine Learning Algorithm

9.1.1 Decision Tree

According to Tables 2 and 3, the Decision Tree (DT) algorithm performs well in predicting benign and malware URLs. However, its performance slightly diminishes when identifying defacement URLs, and it shows very low performance in predicting phishing URLs. Fig. 9 illustrates the confusion matrix for the validation and testing phases of the DT algorithm.

Table 2: DT validation results

URL type	Precision	Recall	F1 score	Accuracy	Training time (s)	Validation time (s)
0	0.85	0.85	0.85	0.78	7.22	0.49
1	0.72	0.72	0.72			
2	0.85	0.86	0.86			
3	0.52	0.51	0.51			

Table 3: DT testing results

URL type	Precision	Recall	F1 score	Accuracy	Testing time (s)
0	0.84	0.86	0.85	0.78	0.33
1	0.73	0.72	0.72		
2	0.85	0.86	0.86		
3	0.52	0.49	0.51		

The results from both validation and testing phases are almost similar, indicating that the model is not overfitting.

9.1.2 Naïve Bayes

Based on Tables 4 and 5, Naïve Bayes achieves an accuracy of 68% in both the validation and testing datasets. However, there are notable differences in its effectiveness in predicting specific types of URLs. The model performs well in predicting benign URLs (Type 0), but its performance decreases significantly for defacement (Type 1), malware (Type 2), and phishing (Type 3) URLs, especially in terms of recall. This suggests that the Naïve Bayes algorithm may fail to identify certain potentially harmful types of URLs.

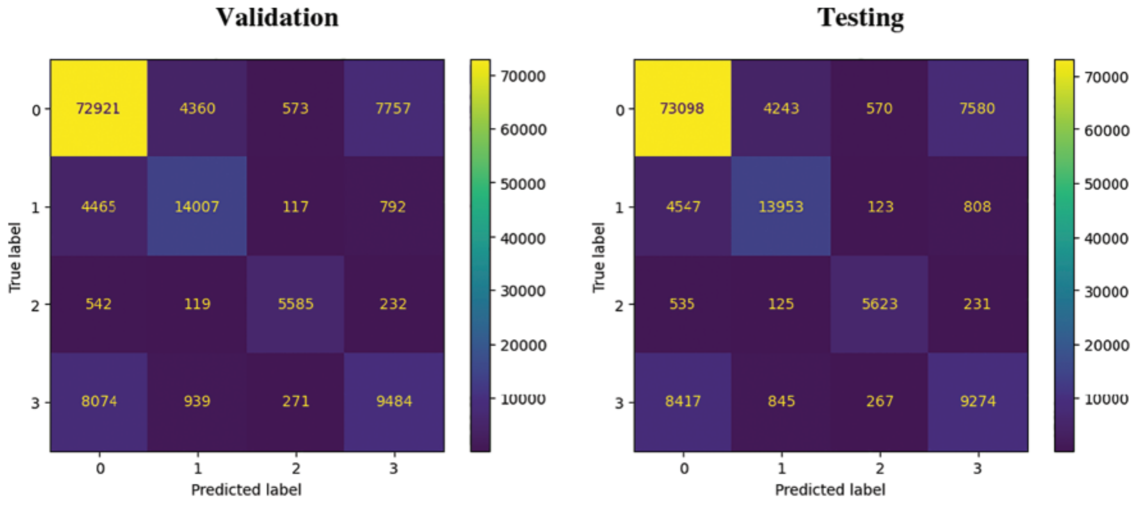


Figure 9: Confusion matrix during validation and testing

Table 4: NB validation results

URL type	Precision	Recall	F1 score	Accuracy	Training time (s)	Validation time (s)
0	0.81	0.76	0.78	0.68	0.29	0.57
1	0.45	0.63	0.52			
2	0.41	0.61	0.49			
3	0.56	0.39	0.46			

Table 5: NB testing results

URL type	Precision	Recall	F1 score	Accuracy	Testing time (s)
0	0.81	0.76	0.78	0.68	0.41
1	0.45	0.63	0.52		
2	0.42	0.62	0.50		
3	0.56	0.39	0.46		

The confusion matrix in Fig. 10 illustrates that the Naïve Bayes algorithm incorrectly labels defacement, malware, and phishing URLs during testing. These results indicate that Naïve Bayes is less efficient compared to the Decision Tree algorithm.

9.1.3 Random Forest

The above results demonstrate the performance of the Random Forest (RF) algorithm in both validation and testing phases. Tables 6 and 7 indicate that the RF model performs exceptionally well across all four types of URLs, achieving a high accuracy of 87% in both validation and testing.

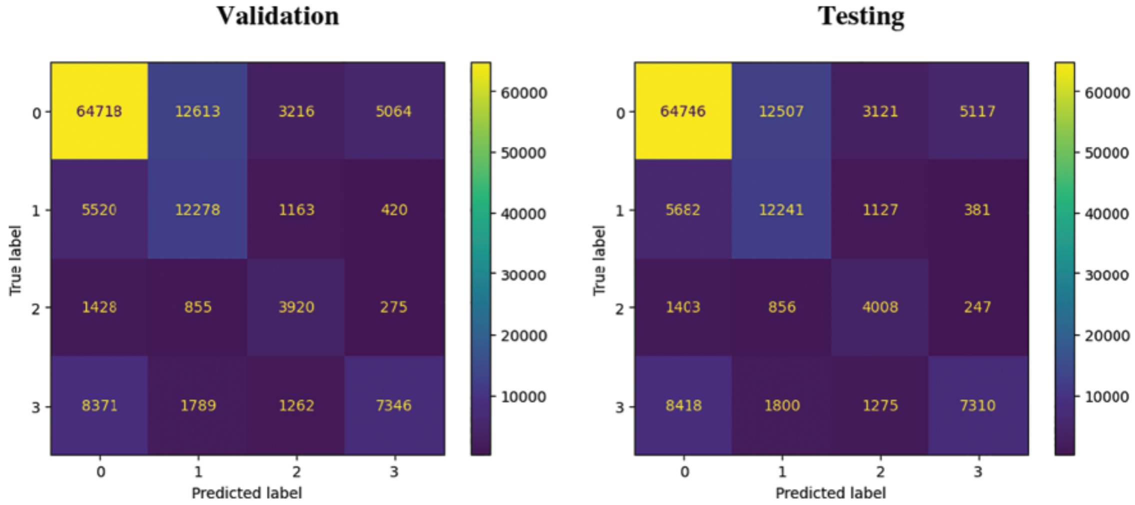


Figure 10: Confusion matrix during validation and testing

Table 6: RF validation results

URL type	Precision	Recall	F1 score	Accuracy	Training time (s)	Validation time (s)
0	0.85	0.98	0.91			
1	0.96	0.73	0.83	0.87	1238.28	18.86
2	0.99	0.88	0.93			
3	0.88	0.49	0.63			

Table 7: RF testing results

URL type	Precision	Recall	F1 score	Accuracy	Testing time (s)
0	0.85	0.98	0.91		
1	0.96	0.72	0.83	0.87	24.76
2	0.99	0.88	0.93		
3	0.87	0.49	0.63		

Specifically, the RF model exhibits strong performance in predicting defacement URLs, achieving a precision of 96%, recall rate of 73%, and F1 score of 83%. This can be because of the default characteristics found in the defacement URLs.

For malware URLs, the RF model achieves an impressive F1 score of 93%, recall score of 88%, and precision score of 99%.

Fig. 11 presents the confusion matrix for the RF algorithm in both validation and testing phases. Despite its overall strong performance, the model shows slightly lower effectiveness in predicting benign and phishing URLs, as observed from Tables 6 and 7. Nonetheless, Random Forest proves to be effective in identifying harmful URLs.

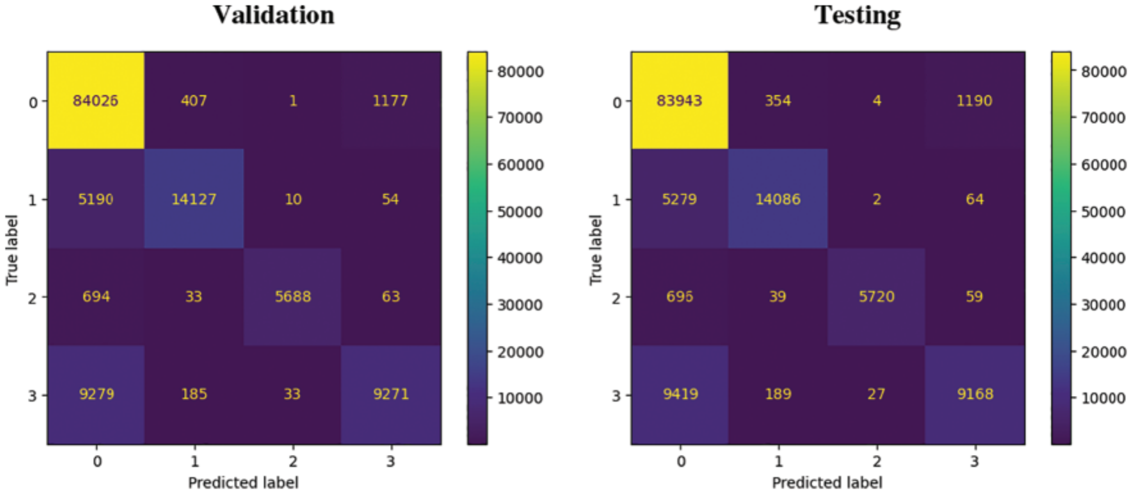


Figure 11: Confusion matrix during validation and testing

9.2 Deep Learning

9.2.1 Multi-Layer Perceptron

Based on Tables 8 and 9, the Multilayer Perceptron (MLP) demonstrates strong performance in predicting malicious URLs with an accuracy of 82% in both the validation and testing datasets. However, there are variations in its effectiveness across different types of URLs. The model performs well (over 80%) in predicting benign, defacement, and malware URLs, but its performance is slightly lower when predicting phishing URLs.

Table 8: MLP validation results

URL type	Precision	Recall	F1 score	Accuracy	Training time (s)	Validation time (s)
0	0.82	0.95	0.88	0.82	975.76	0.84
1	0.82	0.67	0.74			
2	0.87	0.78	0.82			
3	0.75	0.40	0.52			

Table 9: MLP testing results

URL type	Precision	Recall	F1 score	Accuracy	Testing time (s)
0	0.82	0.95	0.88	0.82	0.79
1	0.82	0.66	0.73		
2	0.86	0.79	0.83		
3	0.75	0.40	0.52		

The confusion matrix in Fig. 12 indicates that there is a possibility for the model to misclassify phishing URLs as benign. This could be due to common characteristics between phishing and benign URLs.

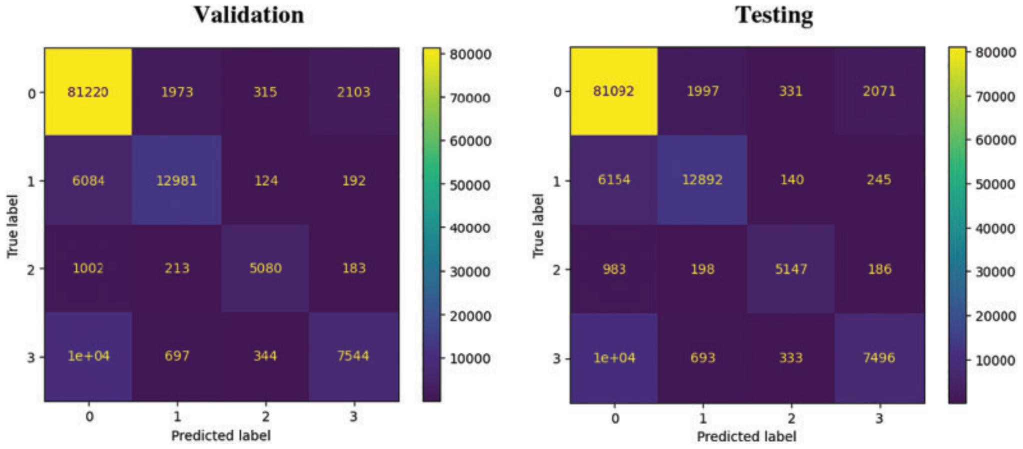


Figure 12: Confusion matrix during validation and testing

MLP exhibits a training time of 975.76 s and a testing time of 0.79 s, which are relatively longer. Despite this, the findings suggest that MLP is an efficient algorithm for predicting malicious URLs. It's worth noting that Multilayer Perceptron (DL) requires less training and testing time compared to Random Forest (ML).

9.3 Performance of Individual Algorithms and ML-DL Combined Algorithms

Table 10 presents the performance of different combinations of machine learning (ML) and deep learning (DL). The following combinations in Table 10 achieve a testing accuracy of 80%: (a) DT, NB, RF & MLP; (b) DT, RF & MLP; (c) RF & MLP. The RF and MLP combination exhibits a training time of 33.78 s, a validation time of 9.61 s, and a testing time of 9.41 s.

Table 10: Results of individual algorithms and ML-DL combined algorithms

Individual/Combined	Algorithms	Accuracy		Time taken		
		Validation	Testing	Training	Validation	Testing
Individual	DT (ML)	0.78	0.78	7.22	0.49	0.33
	NB (ML)	0.68	0.68	0.29	0.57	0.41
	RF (ML)	0.87	0.87	1238.28	18.86	24.76
	MLP (DL)	0.82	0.82	975.76	0.84	0.79
Combined	RF & MLP	0.81	0.80	33.78	9.61	9.41
	MLP & DT	0.77	0.77	17.05	1.19	1.20
	NB & MLP	0.78	0.78	16.08	1.30	2.04
	DT, NB, RF & MLP	0.80	0.80	33.88	10.61	13.69
	NB, RF & MLP	0.79	0.79	35.80	9.26	10.37
	DT, NB & MLP	0.78	0.78	16.27	1.35	1.33
	DT, RF & MLP	0.80	0.80	36.12	10.00	9.79

Although combinations like MLP & DT, NB & MLP, NB, RF & MLP, and DT, NB, & MLP may have slightly lower accuracy, they offer quicker validation, training, and testing times.

The following combination has the best performance, RF & MLP, NB, RF & MLP, and DT, RF & MLP. However, if speed is crucial, the RF & MLP combination stands out as the most effective in identifying malicious URLs.

10 Conclusion

This work focused on developing the best model for predicting malicious URLs using machine learning and deep learning. Three machine learning algorithms, namely Decision Tree, Random Forest, and Naïve Bayes, and one deep learning algorithm, namely Multilayer Perceptron, were used in this paper. All the machine learning and deep learning algorithms were evaluated individually and in different combinations using various evaluation metrics. The combination of Random Forest (RF) and Multilayer Perceptron (MLP) was found to be the best model with an accuracy of 81%. It balances accuracy, training time, and testing time, making this combination the most efficient among all the algorithms.

Acknowledgement: None.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: Study, conceptualization and design, Imran Khan, Meenakshi Megavarnam; Data collection, Meenakshi Megavarnam; Analysis and interpretation results, Imran Khan; Supervision, Imran Khan; Manuscript preparation, Meenakshi Megavarnam. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The authors confirm that the data supporting the findings of this study are available within the article.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

[1] L. Andrew, "The vulnerability of vital systems: How 'critical infrastructure' became a security problem," in *Securing 'the Homeland'*. London, UK: Routledge, 2020, pp. 17–39.

[2] C. Iwendi, J. H. Anajemba, C. Biamba, and D. Ngubo, "Security of things intrusion detection system for smart healthcare," *Electronics*, vol. 10, no. 12, Jun. 2021, Art. no. 1375. doi: 10.3390/electronics10121375.

[3] X. Wu *et al.*, "Threat analysis for space information network based on network security attributes: A review," *Complex Intell. Syst.*, vol. 9, no. 3, pp. 3429–3468, Nov. 2022. doi: 10.1007/s40747-022-00899-z.

[4] L. Tan, N. Shi, K. Yu, M. Aloqaily, and Y. Jararweh, "A chain-empowered access control framework for smart devices in green internet of things," *ACM Trans. Internet Technol.*, vol. 21, no. 3, pp. 1–20, Jun. 2021.

[5] F. Khan, J. Ahamed, S. Kadry, and L. K. Ramasamy, "Detecting malicious URLs using binary classification through adaboost algorithm," *Int. J. Electr. Comput. Eng.*, vol. 10, no. 1, pp. 997–1005, Feb. 2020. doi: 10.11591/ijece.v10i1.pp997-1005.

[6] M. K. Prabakaran, P. Meenakshi Sundaram, and A. D. Chandrasekar, "An enhanced deep learning-based phishing detection mechanism to effectively identify malicious URLs using variational autoencoders," *IET Inf. Secur.*, vol. 17, no. 3, pp. 423–440, Jan. 2023.

[7] M. M. Aljabri *et al.*, "An assessment of lexical, network, and content-based features for detecting malicious urls using machine learning and deep learning models," *Comput. Intell. Neurosci.*, vol. 2022, no. 1, 2022, Art. no. 3241216. doi: 10.1155/2022/3241216.

[8] W. Yang, W. Zuo, and B. Cui, "Detecting malicious URLs via a keyword-based convolutional gated-recurrent-unit neural network," *IEEE Access*, vol. 7, pp. 29891–29900, Jan. 2019. doi: 10.1109/ACCESS.2019.2895751.

[9] O. V. Lee *et al.*, "A malicious URLs detection system using optimization and machine learning classifiers," *Indones J. Electr. Eng. Comput. Sci.*, vol. 17, no. 3, pp. 1210–1214, Mar. 2020. doi: 10.11591/ijeecs.v17.i3.pp1210-1214.

[10] N. Bhadouria, "Malicious_URL's_Dataset," 2022. Accessed: Jun. 5, 2023. [Online]. Available: https://www.kaggle.com/datasets/naveenbhadouria/malicious

[11] A. Alanazi and A. Gumaei, "A decision-fusion-based ensemble approach for malicious websites detection," *Appl. Sci.*, vol. 13, no. 18, Sep. 2023, Art. no. 10260. doi: 10.3390/app131810260.

[12] T. Le, M. Y. Lee, J. R. Park, and S. W. Baik, "Oversampling techniques for bankruptcy prediction: Novel features from a transaction dataset," *Symmetry*, vol. 10, no. 4, Mar. 2018, Art. no. 79.

[13] A. Anagnostis *et al.*, "A deep learning approach for anthracnose infected trees classification in walnut orchards," *Comput. Electron. Agric.*, vol. 182, Mar. 2021, Art. no. 105998. doi: 10.1016/j.compag.2021.105998.

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

[15] Y. Mourtaji, M. Bouhorma, D. Alghazzawi, G. Aldabbagh, and A. Alghamdi, "Hybrid rule-based solution for phishing URL detection using convolutional neural network," *Wirel. Commun. Mob. Comput.*, vol. 2021, no. 1, pp. 1–24, 2021, Art. no. 8241104. doi: 10.1155/2021/8241104.

[16] R. Bayraktar, B. Haznedar, K. S. Bayram, and M. F. Hasoğlu, "Plant disease detection by using adaptive neuro-fuzzy inference system," *Tamap J. Eng.*, vol. 2021, no. 125, pp. 1–10, Sep. 2021.

[17] H. Liu and B. Lang, "Machine learning and deep learning methods for intrusion detection systems: A survey," *Appl. Sci.*, vol. 9, no. 20, Oct. 2019, Art. no. 4396.

[18] A. R. Mohammed, S. A. Mohammed, and S. Shirmohammadi, "Machine learning and deep learning based traffic classification and prediction in software defined networking," in *IEEE Int. Symp. Meas. Netw. (M&N)*, Catania, Italy, Jul. 8–10, 2019, pp. 1–6.