НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ
март–апрель 2023 Том 23 № 2 http://ntv.ifmo.ru/
SCIENTIFIC AND TECHNICAL JOURNAL OF INFORMATION TECHNOLOGIES, MECHANICS AND OPTICS
March–April 2023 Vol. 23 No 2 http://ntv.ifmo.ru/en/
ISSN 2226-1494 (print) ISSN 2500-0373 (online)

НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ
март–апрель 2023 Том 23 Номер 2

doi: 10.17586/2226-1494-2023-23-2-304-312

Natural language based malicious domain detection using machine learning and deep learning

Abdul S. Saleem Raja[1]✉, Ganesan Pradeepa[2], Somasundaram Mahalakshmi[3], Manickam S. Jayakumar[4]

[1,2,4] University of Technology and Applied Sciences, Shinas, 324, Oman
[3] Vivekananda College of Arts and Sciences for Women, Tiruchengode, 637211, India

[1] asaleemrajasec@gmail.com✉, https://orcid.org/0000-0002-7203-1426
[2] pradeepa25.ganesan@gmail.com, https://orcid.org/0000-0002-5920-066X
[3] mahalakshmimsccs@gmail.com, https://orcid.org/0009-0008-5059-4384
[4] jaikumarmanickam@gmail.com, https://orcid.org/0000-0002-5417-5960

Abstract

Cyberattacks are still challenging since they are increasing day by day. Cybercriminals employ a variety of strategies to manipulate and exploit their targets vulnerabilities. Malicious URLs are one such strategy which is used to target large groups on various social media platforms. To draw internet users, these web addresses are disguised as being safe. Deliberate or inadvertent use of such URLs exposes the user or the organization in the cyberspace and opens the way for further attacks. Systems that use rules-based or machine learning algorithms to find malicious URLs usually rely on feature engineering. This requires domain expertise and experience. Sometimes, even after extracting features from a dataset, it may not completely leverage the potential of the dataset. The proposed method employs Natural Language Processing (NLP) approaches to vectorize the words in the URLs and applies machine learning and deep learning models for classification. Vectorization technique in NLP reduces the effort of feature engineering and maximizing the use of the dataset. For the experiment, two separate datasets are used. To vectorize the URL text, three different vectorization methods are used. To evaluate the performance of the proposed method, two different datasets (D1 and D2) that are regularly utilized in the research domain were used. The results demonstrate that the superior accuracy of 92.4 % with the D1 dataset is achieved by the Decision Tree (DT) with count vectorizer and the Random Forest (RF) with Term Frequency-Inverse Document Frequency (TF-IDF) vectorizer. With the D2 dataset, DT with TF-IDF vectorizer obtains a greater accuracy of 99.5 %. The Artificial Neural Network (ANN) model achieves 89.6 % accuracy with the D1 dataset and 99.2 % accuracy with the D2 dataset.

Keywords

malicious domain, phishing URL, NLP, machine learning, deep learning, ANN, CNN

For citation: Saleem Raja A.S., Pradeepa G., Mahalakshmi S., Jayakumar M.S. Natural language based malicious domain detection using machine learning and deep learning. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2023, vol. 23, no. 2, pp. 304–312. doi: 10.17586/2226-1494-2023-23-2-304-312

УДК 004.7

Обнаружение вредоносного домена на основе естественного языка с использованием машинного обучения и глубокого обучения

Абдул Самад Салим Раджа[1]✉, Ганесан Прадипа[2], Сомасундарам Махалакшми[3], Маникам Сам Джаякумар[4]

[1,2,4] Университет технологий и прикладных наук, Шинас, 324, Оман
[3] Колледж искусств и наук Вивекананды для женщин, Тирученгоде, 637211, Индия

[1] asaleemrajasec@gmail.com✉, https://orcid.org/0000-0002-7203-1426
[2] pradeepa25.ganesan@gmail.com, https://orcid.org/0000-0002-5920-066X
3 mahalakshmimsccs@gmail.com, https://orcid.org/0009-0008-5059-4384
4 jaikumarmanickam@gmail.com, https://orcid.org/0000-0002-5417-5960

Аннотация

В настоящие время количество кибератак постоянно увеличивается, и борьба с ними остается сложной задачей. Киберпреступники используют различные стратегии для манипулирования и использования уязвимостей своих целей. Вредоносные URL-адреса — одна из таких стратегий, которая ориентирована на большие группы пользователей, находящихся в социальных сетях. В Интернете для привлечения пользователей преступники маскируют URL-адреса под безопасные. Преднамеренное или непреднамеренное использование таких URL-адресов подвергает опасности пользователя или организацию в киберпространстве и открывает путь для дальнейших атак. Системы, которые используют алгоритмы на основе правил или машинного обучения для поиска вредоносных URL-адресов, обычно полагаются на применение специальных функционалов. Это требует знания предметной области и опыта. Вместе с тем даже при извлечении опасных признаков из набора данных их потенциал может быть применен не полностью. В работе предложено использовать обработку естественного языка (Natural Language Processing, NLP) для векторизации слов в URL-адресах, а также моделей машинного и глубокого обучения для их классификации. Техника векторизации при обработке естественного языка позволяет снизить усилия по разработке признаков и максимально использует набор данных. Для эксперимента применены два набора данных, а для векторизации текста URL — три метода. Результаты эксперимента показали, что модели дерева решений (Decision Tree, DT) и метода случайного леса (Random Forest, RF) достигли точностей 99,4 % и 99,3 % с использованием машинного обучения с векторизаторами Count и Hash. Модели DT и метода опорных векторов (Support Vector Machine, SVM) обеспечили высокую точность 99,5 % с использованием меры Term Frequency-Inverse Document Frequency (TF-IDF). В модели глубокого обучения нейронной сети (Artificial Neural Network, ANN) получена точность 99,2 %, что выше в сравнении с использованием сверточной нейронной сети (Convolutional Neural Network, CNN).

Ключевые слова

вредоносный домен, фишинговый URL, NLP, машинное обучение, глубокое обучение

Ссылка для цитирования: Салим Раджа А.С., Прадипа Г., Махалакшми С., Джаякумар М.С. Обнаружение вредоносного домена на основе естественного языка с использованием машинного обучения и глубокого обучения // Научно-технический вестник информационных технологий, механики и оптики. 2023. Т. 23, № 2. С. 304–312 (на англ. яз.). doi: 10.17586/2226-1494-2023-23-2-304-312

Introduction

In the 21st century, internet and world wide web offers variety of service to the users and makes the life easier ever before. But it also brings many cybersecurity threats, such as malwares, data breaching, spamming, financial frauds, etc. Every year, the cybercrime rate is increasing exponentially and becoming a serious obstacle to digitalization[1]. This makes cybersecurity a major concern in today's business world. Especially in covid-19 pandemic period, companies, organizations, and institutions are switching to online mode, which encourages cybercriminals to increase their attacks in numbers through different mediums. Recent security report states that business companies around the world spends $123 billion in 2020 and $170.4 billion in 2021[2]. Cyber criminals use different tactics to deceive the people to achieve their objective. Malicious URLs are the primary tactic for them to create mass attack. The URL of the malicious websites mimic legitimate URL with catchy words. Most of the time malicious URLs are spread through SMS or social media and injected on legitimate websites. Deliberate or inadvertent use of such URLs exposes the user or organization in cyberspace and opens the way for further action. Once the user clicks on those links, it downloads malware in the user's system or steal sensitive information from the user's system, exposing the user's system vulnerabilities in cyberspace. Detecting malicious URLs in cyberspace is a daunting task for the user. To prevent such links, the security software follows a blacklisting approach that does not allow access to the URLs in the blacklist. There is a large list of URLs in the blacklist which are created manually or automatically by the system. Maintaining and updating such a blacklist is a challenging task. Timely update is essential otherwise newly generated malicious URLs are easily evading the security system [1]. Another approach is based on heuristic rules, it begins by extracting essential features from URLs (both blacklist and whitelist) and creating general rules for the detection system. This approach is better than the blacklist approach. However, generalizing the rules for detection requires extensive work [2]. Third approach is machine learning based detection using the features that are extracted from different sources such as URLs, content of the websites, information from the servers and visual similarity of the websites [3]. The performance of a machine learning model-based system depends on the feature engineering which include feature extraction, selection, scaling, etc. The process of feature engineering requires expertise knowledge in the domain. Moreover, selected number of features may not exploit the potential of the huge dataset [4]. To overcome these issues, researchers prefer to use NLP techniques to leverage the potential of the dataset. Our research work uses different NLP technique to vectorize the URL text instead extracting conventional features from the URLs and tested with different machine learning and deep learning algorithms for malicious URL detection.

Contribution of our proposed work:

— Processing only URL text which is more safe method than the content processing of the webpage.

— Using NLP techniques for URL text processing simplifies the processing overhead of feature engineering.

[1] Cybersecurity. Available online: https://www.forbes.com/sites/louiscolumbus/2020/08/09/cybersecurity-spending-to-reach-123b-in-2020/?sh=5f107e56705f (accessed: 18.10.2022).

[2] Cybersecurity Statistics. Available online: https://www.fortinet.com/resources/cyberglossary/cybersecurity-statistics (accessed: 18.10.2022).

— The performance of machine learning and deep learning models are evaluated and presented for a malicious webpage detection system based on the URL text.

Background

URL is a string and a unique identifier to locate the resource in the cyberspace. It is composed of several elements [5] that are presented in Table 1. Generally, machine learning models operated on numerical features that are extracted from the raw dataset is represented as 2-D array. Most of the existing research work uses the four categories of features for the malicious URL classification [6] as shown in Table 2.

This paper uses NLP technique to vectorize the URL text instead of extracting conventional features. NLP offers set of techniques for vectorization. The process of transforming text data into numerical features is known as vectorization (feature extraction). Most commonly used methods are:

1) Count Vectorizer;
2) Term Frequency–Inverse Document Frequency;
3) Hash Vectorizer;
4) Word Embedding.

Vectorized data is the input to machine learning algorithms for classification. Table 3 listed the machine learning algorithms for our experiment.

Count Vectorizer [7, 8]. It is a simple technique to represent the text data into a numerical vector which counts the number of the occurrences of the word in a document. The output of this technique is a sparse matrix.

Term Frequency–Inverse Document Frequency (TF-IDF) [7, 8]. It is used to reflect the relevance of a word in a corpus. Term Frequency (TF) is defined as the frequency of the word present in a document and it's computed as shown below.

$$\mathrm{TF} = \frac{\text{Number of occurence of the word in a document}}{\text{Total number of the words in the document}}.$$

Inverse Domain Frequency (IDF) is defined as the frequency of the word across the set of documents and it's computed as shown below. IDF indicates the importance of the word. In general, less frequent words are more informative.

$$\mathrm{IDF} = \lg\frac{\text{Total number of document}}{\text{Number of document that contain the word}}.$$

At last, the score of the TF-IDF is computed as shown below

$$\text{TF-IDF} = \mathrm{TF} \times \mathrm{IDF}.$$

Hash Vectorizer [7, 8]. Hash vectorizer is similar to count vectorizier that converts raw text into numerical vector (matrix) but holds less memory because of not storing the resulting vocabulary.

Keras Word Embedding[1] [9]. This technique helps to keep close words that are semantically similar. It transforms words in a vocabulary to dense vector of real numbers.

Deep learning (DL) is a subset of machine learning that has recently been used extensively in the text classification problem in NLP [11]. DL is simply a neural network with set of layers (one input, one or more hidden, one output layers). Each layer consists of nodes called neurons. Every

[1] Word Embedding. Available online: https://machinelearningmastery.com/use-word-embedding-layers-deep-learning-keras/ (accessed: 18.10.2022).

Table 1. Elements of the URL [5]

Elements of URL	Description	Example
Protocol (Scheme)	Protocol is used to access the resources	http
Domain Name	Distinct name of the resources which is translated as actual IP address of the hosted server	www.myspace.com
Port Number	Unique number for every protocol to access the resources	80 for http
Path	Path name of the requested resource on the server. Generally relative location of the folder and file	myfolder/web/page1.html

Parameters	It's a key value pair for access the resource in the server	name=x&
		password=y

Anchor (Fragment)	Content location of the webpage is marked as anchor or fragment	#contentlocation

Table 2. Categories of Features [6]

Category	Description
Features extracted from URL string	Count the various elements in the URL, such as counting number of dot, hyphens, etc.
Features extracted from webpage content	Count the various elements in the webpage, such as counting number of script tags, paragraph tags, etc. Counting specific words in the webpage
Features extracted from Domain name server	IP address, registration date, expiry date, hosted server and other related information of the webpage
Features extracted from third party server	Ranking of webpage from google or alexa, etc.

Table 3. Machine learning algorithms [10]

Algorithm	Description
Logistic Regression (LR)	Used to predict the binary target class based on the independent features. It uses sigmoid function to transform the real value into discrete class
Support Vector Machine (SVM)	Used to determine the hyperplane to classify the data points
Decision Tree (DT)	Construct the tree, based on the input data and are divided based on the conditions in the internal node. It can be used for regression and classification problems
K-Nearest Neighbors (KNN)	Group the data points which are close to the chosen data point. The value K determines number of groups
Random Forest (RF)	Group of DT are created by the algorithm. The output of the algorithm is the mean of prediction of all the decision trees
Gradient Boosting (GB)	Combines several weak models to get better performance model
AdaBoost (AB)	Adaptive boosting combines the weaker model into a single strong model and used for binary classification
XGBoost (XGB)	XGBoost algorithm is an extended version of the gradient boosting algorithm

node in a layer is connected with other nodes in another layer with weight and threshold value. If any node output is above the threshold, value is activated to transfer data to the subsequent layer. For our experiment, both machine learning and deep learning models Artificial Neural Network (ANN), Convolutional Neural Network (CNN) are used to classify the malicious URLs along and results are presented.

Related work

Malicious URL includes phishing, spamming, defacement, and malware URLs. Researchers more preferably use URL string and text content of the malicious webpage for classification. Text processing in NLP can be applied to both URL string as well as content of the webpage. Lakshmanarao et al. [12] proposed a malicious website detection system which uses NLP techniques (count vectorizer, TF-IDF vectorizer and hash vectorizer) for feature extraction. Different machine learning based classifiers were used for testing the Kaggle dataset and the result shows that RF algorithm gives 97.5 % accuracy. Bocheng et al. [13] presented a deep learning based malicious detection system using NLP. Experiments were conducted with machine and deep learning algorithms. LSTM model with attention mechanism achieves 98.9 % accuracy. Routhu et al. [14] proposed a phishing detection method by using text of webpages. 10,000 URLs including phishing and benign were used for the experiment. Result of the experiments shows that the multimodel domain specific text gives 99.34 % detection accuracy. Zhang et al. [15] presented a system which extracts different features from URLs as well as content, visual similarity, reputation based features, and applying NLP technique for text processing. The result shows that adaboost plus word embedding method gives 99.8 % accuracy. Malak et al. [16] presented a phishing detection system using machine and deep learning. Two different datasets (UCI Phishing dataset and Kaggle dataset) were used to extract hybrid feature. Set of ML and DL models were used for the experiment. The results show that the random forest algorithm offers 96.53 % accuracy. Eint et al. [17] proposed phishing URL detection method by using URL's domain and path features. Performance of the proposed method is evaluated by using balanced (D1) and imbalanced (D2) dataset. Keras and Embeddings from Language Models word embedding is used to encode the text. The result shows LSTM network gives accuracy of 97 % with D2.

The majority of the works that are currently available use a hybrid feature set for malicious web page detection, which combines the conventional features of URL text with features taken from the content of the web page. The process of extracting and selecting features is complex, time-consuming, and calls for prior knowledge of the relevant domain. Moreover, the selected features may not make use of the dataset's full potential. Processing web page content can sometimes be harmful to the system. To solve these problems, the proposed system uses only URL text for experiments and uses NLP techniques to vectorize the URL text, which gets rid of the need to extract and to choose features.

Proposed Method

The aim of the research work is to detect the malicious URLs by utilizing the URL of the web pages. Proposed method uses NLP techniques to vectorize the URL text, and different machine learning & deep learning models are evaluated for better detection accuracy. The function of the proposed work is depicted in Fig. 1.

For the purpose of evaluating the machine learning and deep learning models, two separate datasets (D1 & D2) are used as shown in Table 4. Preprocessing is the first step that identifies the required data from the raw dataset. In our case, only the URLs of the webpages are used. For the detection process, the domain names are taken from the URL because they have more important information than the other parts of the URL. After domain names are retrieved and sanitized, vectoriation converts text data into numerical features. Finally, the dataset is separated into training and testing sets for machine learning and deep learning models. Dataset for the experiment consists

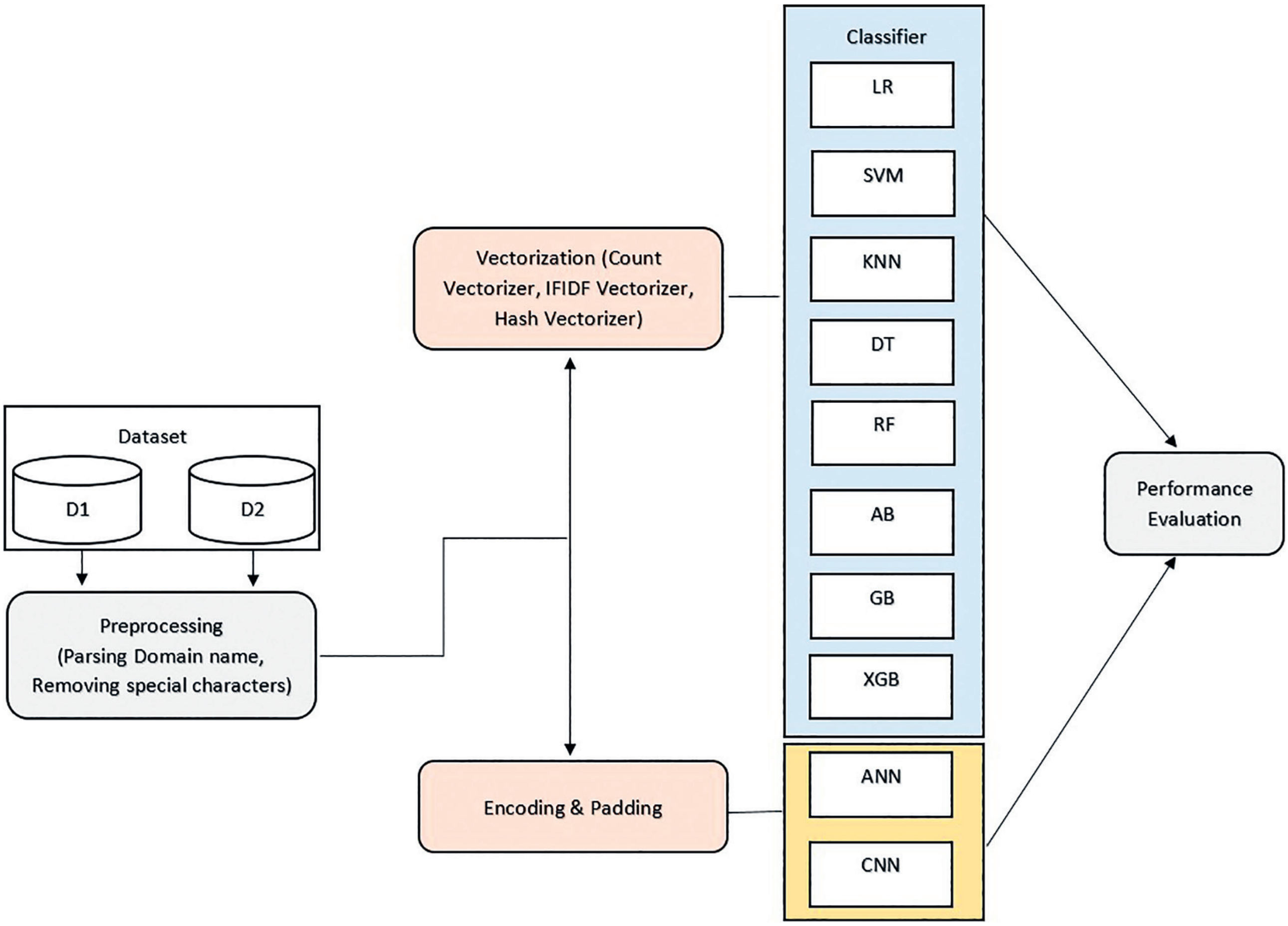
Dataset
D1
D2
Preprocessing (Parsing Domain name, Removing special characters)
Vectorization (Count Vectorizer, IFIDF Vectorizer, Hash Vectorizer)
Encoding & Padding
Classifier
LR
SVM
KNN
DT
RF
AB
GB
XGB
ANN
CNN
Performance Evaluation

Fig. 1. Process workflow of our proposed work

Table 4. Dataset for the experiment

No	Source	Type of URL	Total number of URL	Dataset
1	URL dataset (ISCX-URL2016)	Malicious URL	5000	D1
		Benign URL	5000	
2	UNB	Benign URL	5000	D2
	Phishtank	Phishing URL	5000	

of URLs collected from URL dataset (ISCX-URL2016)[1] (includes spam, malware, phishing, defacement and benign URLs), UNB[2] (benign URLs), and phistank[3] (phishing URLs). Except the benign URLs, all other URLs (spam, malware, defacement, phishing) are considered as malicious URLs for the experiment.

During pre-processing, the domain names are extrated from the URLs. This is because the domain name of the URL provides more important information for the classification than the rest of the URL [18]. So our experiment only considers the domain names of URLs. After parsing domain names, the next step is to remove unwanted special characters and clean them up. Vectorization is a technique in NLP to transform the text data into vector of real number which is suited for machine learning model. Different vectorization techniques (Bag of words (Count Vectorizer), TF-IDF Hash Vectorizer, Word Embedding) are available in NLP. Our proposed work utilizes all the techniques, and its performances are analyzed. As part of encoding and decoding, text data are converted into numerics by using encoding techniques (integer encoding and one hot encoding) before processing them using machine or deep learning model. For our experiments, one hot encoding technique is used. Moreover, it also requires to have the inputs that are same in size and shape. Padding method is used to make the input data of same size and shape. The performance of the classifier is analyzed using machine learning and deep learning models different metrics are used to evaluate the performance of the model. The parameters that are used to analyze the

[1] Malicious URLs dataset. Available online: https://www.kaggle.com/datasets/sid321axn/malicious-urls-dataset?resource=download (accessed: 18.10.2022).

[2] URL dataset. Available online: https://www.unb.ca/cic/datasets/url-2016.html (accessed: 18.10.2022).

[3] PhishTank. Available online: https://www.phishtank.com/developer_info.php (accessed: 18.10.2022).

performance of the classifier are Accuracy, Precision, Recall, and F-Score. TP is True positive which indicates the total number of correctly predicted positive instances. TN is true negative which indicates the totoal number of negative instances that are predicted as negative. FP is false positive which indicates the total number of negative instances predicted as positive. FN is false negative which indicates the total number of negative instances that are predicted as negative.

Accuracy, Precision, Recall and F1-Score are the other performance metrics tested with the proposed work.

Experiment and result

Experiments are conducted in the windows 10 system with i5 processor running at 3.2 GHz speed, and 8 GB of RAM. Juypter notebook and Python with sklearn, nltk, genism packages are used for programming. To train and test the machine learning and deep learning models, the dataset is divided into training set and testing set in which 80 % of the data is utilized to train the model and 20 % data is preserved for test. Domain names in the URLs are preprocessed and applying different vectorizers to test with the machine learning models. Encoding and padding are done before utilizing the input with deep learning model. Once after the raw text in URLs are converted into padded sequence of same length, it will be mapped to an array of real numbers by word embedding method which is the first hidden layer in neural network. The deep learning models configuration are given in the Table 5.

Evaluating the performance of models with D1. URL (ISCX-URL2016) dataset is used for the experiment. The result of the experiment is presented in the below figures. The specialty of URL (ISCX-URL2016) is that it contains all sorts of URLs including spam, distortion, malware, phishing, and malicious URLs. Moreover, dataset is a balanced one.

Fig. 2 shows the performance comparison of difference classifiers in terms of different performance metrics. The results show that DT and RF models achieve greater accuracy in machine learning using Count vectorizer and Hash vectorizer techniques. RF and SVM models provide high accuracy using the TF-IDF vectorizer. In the deep learning model, the ANN achieves better accuracy than the CNN.

Evaluating the performance of models with D2. UNB and Phishtank dataset is used for the experiment. The result of the experiment is presented in the below figures. Phishing is the most common type of attack in cyberspace. Therefore, D2 adds weightage to the phishing URLs. Also, the dataset is balanced.

Fig. 3 shows the performance comparison of difference classifiers in terms of different performance metrics. The results show that DT and RF models achieve greater

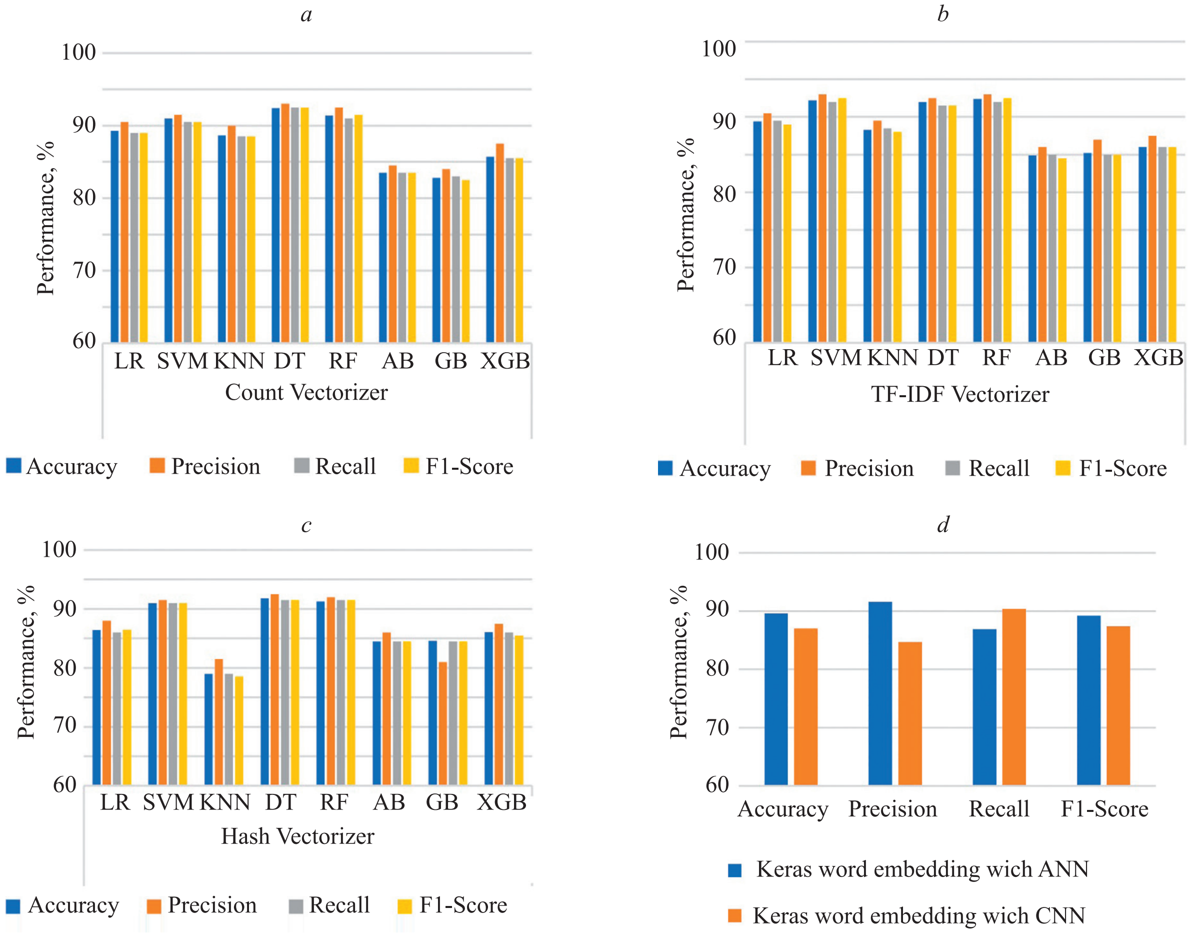

Fig. 2. Performance comparison of various models with dataset D1: Count Vectorizer (*a*), TF-IDF Vectorizer (*b*), Hash Vectorizer (*c*), ANN vs. CNN (*d*)

Table 5. Deep learning model parameters

Model	Parameter	Value
ANN	Training and Test Split	80:20
	Keras Embedding Dimension (size of the vocabulary, size of the vector space)	10,000.10
	Activation function	Relu, Sigmoid
	No. of nodes in the output layer	1
	Batch size	64
	Loss function	Binary cross entropy
	Optimizer	Adam
	Epoch size	10
CNN	Training and Test Split	80:20
	Keras Embedding Dimension (size of the vocabulary, size of the vector space)	10,000.10
	Conv1D	Filter =32,kernel=8
	Pool Size	2
	Activation function	Relu, Sigmoid
	No. of nodes in the output layer	1
	Batch size	64
	Loss function	Binary cross entropy
	Optimizer	Adam
	Epoch size	10

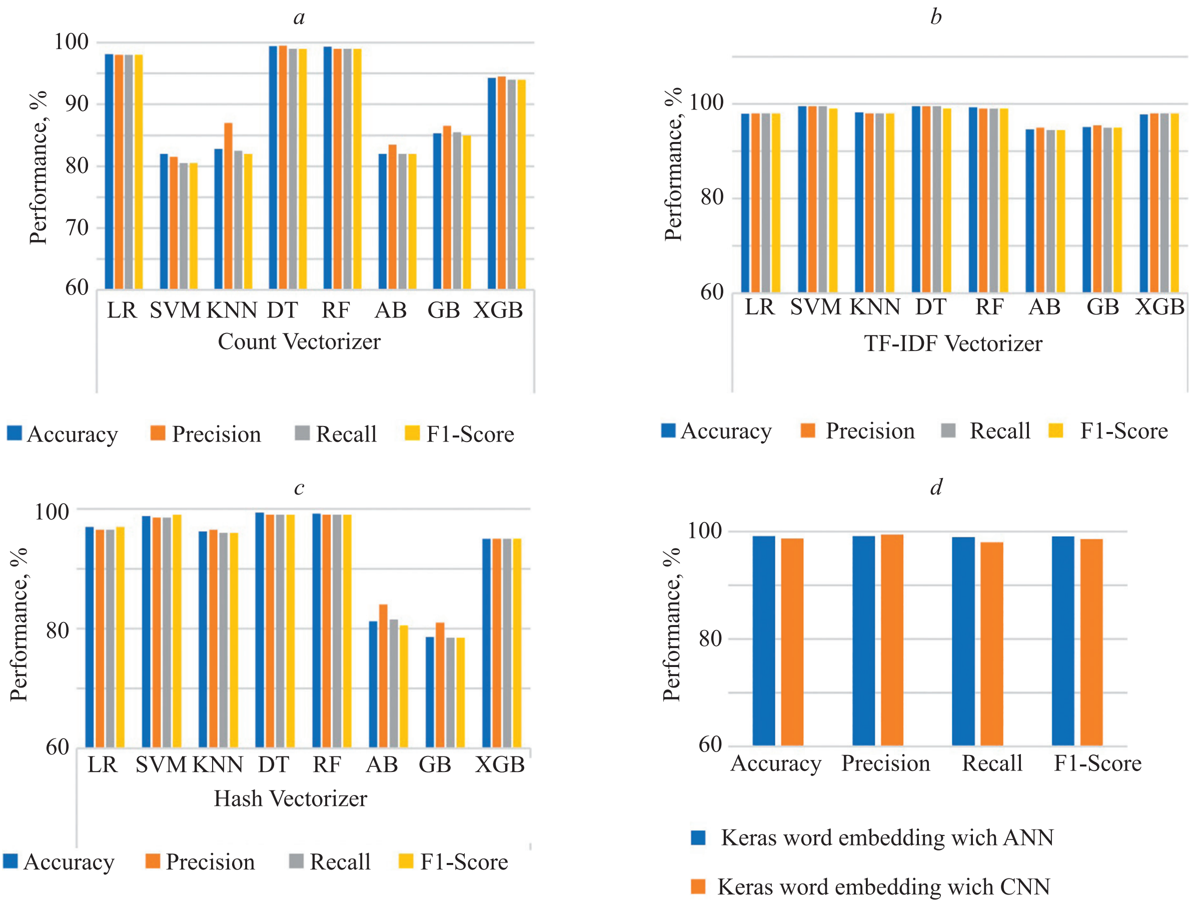

Fig. 3. Performance comparison of various models with dataset D2: Count Vectorizer (*a*), TF-IDF Vectorizer (*b*), Hash Vectorizer (*c*), ANN vs. CNN (*d*)

accuracy in machine learning using Count vectorizer and Hash vectorizer. DT and SVM models provide high accuracy using the TF-IDF vectorizer. In deep learning model, ANN model achieves the better accuracy than the CNN model.

Conclusion

Malicious websites are the convenient way for cybercriminals to commit illegal acts. Numerous studies have already been done by research scholars to detect malicious webpage. Nevertheless, the problem remains open due to the ever-changing attack mode. Some researchers use features based on the content of the webpage. However, this carries the risk of processing dangerous content. Furthermore, domain expertise is required to extract features and sometimes selected features may exploit the potential of the dataset. To address these issues, our proposed work uses only URLs and NLP based text encoding techniques that helps to reduce the feature engineering work as well as utilize the potential of the dataset. The experiment result confirms that the TF-IDF vectorization technique improves the performance of the decision tree and random forest machine learning algorithms with respect to both datasets. Deep learning algorithm uses word embedding techniques, and the results reveal that ANN performs well with both datasets.

References

1. Da H., Xu K., Pei J. Malicious URL detection by dynamically mining patterns without pre-defined elements. *World Wide Web*, 2014, vol. 17, no. 6, pp. 1375–1394. https://doi.org/10.1007/s11280-013-0250-4
2. Saleem Raja A., Pradeepa G., Arulkumar N. Mudhr. Malicious URL detection using heuristic rules based approach. *AIP Conference Proceedings*, 2022, vol. 2393, no. 1, pp. 020176. https://doi.org/10.1063/5.0074077
3. Sahoo D., Liu C., Hoi S.C.H. Malicious URL detection using machine learning: A survey. *ArXiv*, 2017, arXiv:1701.07179. https://doi.org/10.48550/arXiv.1701.07179
4. Brownlee J. *Deep Learning with Python: Develop Deep Learning Models on Theano and TensorFlow Using Keras*. Machine Learning Mastery, 2016, 256 p.
5. Pradeepa G., Devi R. Lightweight approach for malicious domain detection using machine learning. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2022, vol. 22, no. 2, pp. 262–268. https://doi.org/10.17586/2226-1494-2022-22-2-262-268
6. Saleem R.A., Vinodini R., Kavitha A. Lexical features based malicious URL detection using machine learning techniques. *Materials Today: Proceedings*, 2021, vol. 47, part 1, pp. 163–166. https://doi.org/10.1016/j.matpr.2021.04.041
7. Bengfort B., Bilbro R., Ojeda T. *Applied Text Analysis with Python Enabling Language-Aware Data Products with Machine Learning*. O'Reilly Media, Inc, 2018, 332 p.
8. Vishva E.S., Aju D. Phisher fighter: Website phishing detection system based on URL and term frequency-inverse document frequency values. *Journal of Cyber Security and Mobility*, 2022, vol. 11, no. 1, pp. 83–104. https://doi.org/10.13052/jcsm2245-1439.1114
9. Li S., Gong B. Word embedding and text classification based on deep learning methods. *MATEC Web Conference*, 2021, vol. 336, pp. 06022. https://doi.org/10.1051/matecconf/202133606022
10. Géron A. *Hands-On Machine Learning with Scikit-Learn and TensorFlow*. O'Reilly Media, 2017, 574 p.
11. Zhang M. Applications of deep learning in news text classification. *Scientific Programming for Smart Internet of Things*, 2021, vol. 2021, pp. 6095354. https://doi.org/10.1155/2021/6095354
12. Lakshmanarao A., Raja Babu M., Bala Krishna M.M. Malicious URL detection using NLP, machine learning and FLASK. *Proc. of the International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES)*, 2021, pp. 1–4. https://doi.org/10.1109/ICSES52305.2021.9633889
13. Liu B., Zeng X., Dong P. Malicious URL detection system based on LSTM and attention mechanism. *Journal of Physics: Conference Series*, 2021, vol. 2037, no. 1, pp. 012016. https://doi.org/10.1088/1742-6596/2037/1/012016
14. Routhu S.R., Amey U., Alwyn R.P. Application of word embedding and machine learning in detecting phishing websites. *Telecommunication Systems*, 2022, vol. 79, no. 1, pp. 33–45. https://doi.org/10.1007/s11235-021-00850-6
15. Zhang X., Zeng Y., Jin X.-B., Yan Z.-W., Geng G.-G. Boosting the phishing detection performance by semantic analysis. *Proc. of the*

Литература

1. Da H., Xu K., Pei J. Malicious URL detection by dynamically mining patterns without pre-defined elements // World Wide Web. 2014. V. 17. N 6. P. 1375–1394. https://doi.org/10.1007/s11280-013-0250-4
2. Saleem Raja A., Pradeepa G., Arulkumar N. Mudhr. Malicious URL detection using heuristic rules based approach // AIP Conference Proceedings. 2022. V. 2393. N 1. P. 020176. https://doi.org/10.1063/5.0074077
3. Sahoo D., Liu C., Hoi S.C.H. Malicious URL detection using machine learning: A survey // ArXiv. 2017. arXiv:1701.07179. https://doi.org/10.48550/arXiv.1701.07179
4. Brownlee J. Deep Learning with Python: Develop Deep Learning Models on Theano and TensorFlow Using Keras. Machine Learning Mastery, 2016. 256 p.
5. Pradeepa G., Devi R. Lightweight approach for malicious domain detection using machine learning // Научно-технический вестник информационных технологий, механики и оптики. 2022. Т. 22. № 2. С. 262–268. https://doi.org/10.17586/2226-1494-2022-22-2-262-268
6. Saleem R.A., Vinodini R., Kavitha A. Lexical features based malicious URL detection using machine learning techniques // Materials Today: Proceedings. 2021. V. 47. Part 1. P. 163–166. https://doi.org/10.1016/j.matpr.2021.04.041
7. Bengfort B., Bilbro R., Ojeda T. Applied Text Analysis with Python Enabling Language-Aware Data Products with Machine Learning. O'Reilly Media, 2018. 332 p.
8. Vishva E.S., Aju D. Phisher fighter: Website phishing detection system based on URL and term frequency-inverse document frequency values // Journal of Cyber Security and Mobility. 2022. V. 11. N 1. P. 83–104. https://doi.org/10.13052/jcsm2245-1439.1114
9. Li S., Gong B. Word embedding and text classification based on deep learning methods // MATEC Web Conference. 2021. V. 336. P. 06022. https://doi.org/10.1051/matecconf/202133606022
10. Géron A. Hands-On Machine Learning with Scikit-Learn and TensorFlow. O'Reilly Media, 2017. 574 p.
11. Zhang M. Applications of deep learning in news text classification // Scientific Programming for Smart Internet of Things. 2021. V. 2021. P. 6095354. https://doi.org/10.1155/2021/6095354
12. Lakshmanarao A., Raja Babu M., Bala Krishna M.M. Malicious URL detection using NLP, machine learning and FLASK // Proc. of the International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES). 2021. P. 1–4. https://doi.org/10.1109/ICSES52305.2021.9633889
13. Liu B., Zeng X., Dong P. Malicious URL detection system based on LSTM and attention mechanism // Journal of Physics: Conference Series. 2021. V. 2037. N 1. P. 012016. https://doi.org/10.1088/1742-6596/2037/1/012016
14. Routhu S.R., Amey U., Alwyn R.P. Application of word embedding and machine learning in detecting phishing websites // Telecommunication Systems. 2022. V. 79. N 1. P. 33–45. https://doi.org/10.1007/s11235-021-00850-6
15. Zhang X., Zeng Y., Jin X.-B., Yan Z.-W., Geng G.-G. Boosting the phishing detection performance by semantic analysis // Proc. of the

International Conference on Big Data, 2017, pp. 1063–1070. https://doi.org/10.1109/BigData.2017.8258030

16. Malak A., Samitha M. Phishing attacks detection using machine learning and deep learning models. *Proc. of the 7th International Conference on Data Science and Machine Learning Applications (CDMA)*, 2022, pp. 175–180. https://doi.org/10.1109/CDMA54072.2022.00034
17. Aung E.S., Yamana H. Phishing URL detection using information-rich domain and path features. *Proc. of the DEIM*, 2021.
18. Gopinath P., Sangeetha S., Balaji R., Sanjay, Shubham G., Bindhumadhava B.S. Malicious domain detection using machine learning on domain name features, host-based features and web-based features. *Procedia Computer Science*, 2020, vol. 171, pp. 654–661. https://doi.org/10.1016/j.procs.2020.04.071

International Conference on Big Data. 2017. P. 1063–1070. https://doi.org/10.1109/BigData.2017.8258030

16. Malak A., Samitha M. Phishing attacks detection using machine learning and deep learning models // Proc. of the 7th International Conference on Data Science and Machine Learning Applications (CDMA). 2022. P. 175–180. https://doi.org/10.1109/CDMA54072.2022.00034
17. Aung E.S., Yamana H. Phishing URL detection using information-rich domain and path features // Proc. of the DEIM. 2021.
18. Gopinath P., Sangeetha S., Balaji R., Sanjay, Shubham G., Bindhumadhava B.S. Malicious domain detection using machine learning on domain name features, host-based features and web-based features // Procedia Computer Science. 2020. V. 171. P. 654–661. https://doi.org/10.1016/j.procs.2020.04.071

Authors

Abdul Samad Saleem Raja — PhD, Lecturer, University of Technology and Applied Sciences, Shinas, 324, Oman, SC 56862209800, https://orcid.org/0000-0002-7203-1426, asaleemrajasec@gmail.com

Ganesan Pradeepa — Lecturer, University of Technology and Applied Sciences, Shinas, 324, Oman, SC 57673491800, https://orcid.org/0000-0002-5920-066X, pradeepa25.ganesan@gmail.com

Somasundaram Mahalakshmi — Assistant Professor, Vivekananda College of Arts and Sciences for Women, Tiruchengode, 637211, India, https://orcid.org/0009-0008-5059-4384, mahalakshmimsccs@gmail.com

Manickam Sam Jayakumar — Lecturer, University of Technology and Applied Sciences, Шинас, 324, Oman, https://orcid.org/0000-0002-5417-5960, jaikumarmanickam@gmail.com

Авторы

Салим Раджа Абдул Самад — PhD, преподаватель, Университет технологий и прикладных наук, Шинас, 324, Оман, SC 56862209800, https://orcid.org/0000-0002-7203-1426, asaleemrajasec@gmail.com

Прадипа Ганесан — преподаватель, Университет технологий и прикладных наук, Shinas, 324, Оман, SC 57673491800, https://orcid.org/0000-0002-5920-066X, pradeepa25.ganesan@gmail.com

Махалакшми Сомасундарам — доцент, Колледж искусств и наук Вивекананды для женщин, Тиручегоде, 637211, Индия, https://orcid.org/0009-0008-5059-4384, mahalakshmimsccs@gmail.com

Джаякумар Маникам Сам — преподаватель, Университет технологий и прикладных наук, Шинас, 324, Оман, https://orcid.org/0000-0002-5417-5960, jaikumarmanickam@gmail.com

Received 25.11.2022
Approved after reviewing 28.01.2023
Accepted 27.03.2023

Статья поступила в редакцию 25.11.2022
Одобрена после рецензирования 28.01.2023
Принята к печати 27.03.2023