International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Indian Phishing Landscape: A Machine Learning and Deep Learning Approach for Detecting Malicious URLs and Curating an Indigenous Dataset

Dhruv Gada[1], Chinmayee Kale*[2], Himanshu Goswami[3] ,Sridhar Iyer[4]

Submitted:14/03/2024 **Revised**: 29/04/2024 **Accepted**: 06/05/2024

Abstract: Owing to the advancing technologies, activities like browsing the web for critical information, posting on social media and preference for online transactions have increased tremendously. At such a time, it is quite easy to fall into the trap of phishing websites that clone legitimate sites. The paper implements cutting-edge, remarkable Machine Learning techniques which uses Decision Tree algorithm along with Random Forest Classifier, SGD, Logistic Regression, K-Neighbors and Deep Learning methods like CNN, RNN and LSTM on two datasets to classify the online websites as 'phishing' and 'benign'. One of the dataset used is curated to focus on Indian Phishing Website data thus following a more targeted approach in the detection of malicious websites and allowing a proactive defense against attackers. The paper slices and categorizes the data by URL length, top-level domain, symbol counting and other hyper-parameters for seamless feature extraction. The paper also uses TF-IDF for metrics generation as it overcomes limitations of the simple frequency counts and creates a distinctive categorization of the input data values. The fusion of ML and DL techniques to achieve robust cybersecurity, effective training and testing of data on authenticated datasets and a personalized curation of a dataset for top Indian Phishing Websites is how the paper bridges the gap between the research till date.

Keywords: *CNN, Cyber Security, Deep Learning, Indian Malicious Website Dataset, LSTM, Machine Learning, RNN, URL Phishing*

1. Introduction

The internet plays a constitutive role in people's daily roles and activities. It is such an interconnected world, and it is transforming how society communicates, conducts business, and accesses information. Malicious websites, on the other hand, are a pervasive and covert threat in the vast expanse of the online realm.

Malicious websites are a significant risk to individuals and organisations alike and they are purposefully designed to deceive and exploit unsuspecting visitors. Financial losses, identity theft, and even the compromise of sensitive data and systems are all outcomes. Understanding the intricate web of malicious websites is critical in today's digital landscape because it provides users with the knowledge, they need to protect themselves from these threats and navigate the internet safely.

This investigation of malicious websites will delve into the various types and tactics used by cybercriminals, the motivations behind their creation, and the countermeasures that individuals and organisations can use to protect themselves from these hidden threats [8].The paper hopes to illuminate this obscure corner of the internet and provide the world with the awareness and insights it needs to navigate the web with confidence, fortified against the hidden dangers that await with a simple click.

URL phishing is a cyber attack where malicious hackers mimic the appearance of legitimate websites or communication from trusted organisations. In URL Phishing, users are deceived by phished URLs of legitimate websites. The attackers capture and collect the sensitive data entered by the user. The paper, inspired by the research of Wei et al.,[24] takes account of the latest advancements in Machine Learning and Deep Learning in classification and predictive models and implements ML elementary methods like Random Forest Classifier and Stochastic Gradient Descent as well as Decision Tree for better computational efficiency. It also applies a DL algorithm like Convolutional Neural Network (CNN) as it requires less training parameters and to improve the efficiency and precision in detecting a malicious URL [2]. The paper uses python libraries like numpy, pandas, tensor flow and keras to import relevant functions for implementing binary classification.

2. Literature Review

In Rao et al.,[16] study suggests a program called CatchPhish. It introduces hostname, entire length of the URL, Term Frequency-Inverse Document Frequency characteristics, and suspicious terminologies from a dubious URL that predicts the legitimacy of a URL without visiting the website. The only TF-IDF features exhibits an accuracy of 93.25%. When juxtaposed with the prevailing baseline models, the TF-IDF technique to determine the frequency count and handcrafted feature experiment carried out has produced 94.26% accuracy on the dataset used in the paper and 97.49% and 98.25% on alternative standard datasets.

To get around the difficulties in predicting reliable login websites, Paniagua et al., [19] makes advantage of URL, HTML, and web technology properties. The article analysed the authentic login websites to meet the phishing standpoint after creating 134,000 confirmed examples. It uses the full set of 54 features and a LightGBM classifier to obtain an accuracy of 97.95% on the created dataset. The key objective in assessing the legitimacy of a website with a login form was to gauge the paper's ability to replicate an actual phishing detection scenario.

To detect rogue URL addresses with nearly 100% accuracy, Wei et al.,[24] uses convolutional neural networks. Because simply the URL content is analysed, the paper may produce results much more quickly. The network is mobile device optimised, and the paper detects zero-day attacks by machine learning, opposed to blacklist or white-list approaches, making it a mobile-friendly experience in identifying suspicious URLs.

Based on the website's URL, Jiang et al., [1] offers a quick deep learning-based solution model utilising convolutional neural networks (CNN). The suggested method in the study does not call for the retrieval of mediator services or content from the target website. The study applies quick classification of the string and captures sequential patterns of URL strings. Character embedding, TF-IDf, and character level count vector features are only a few of the several feature sets that are used to compare the various machine learning models. On the dataset created for this study, a striking 95.02% accuracy was obtained while it attained accuracies of 95.58%, 95.46%, and 95.22% on the prototype datasets.

Al-Muhtadi et al., [2] proposes three different deep learning-based methods to detect malicious websites: the novel approach LSTM and convolutional neural network (CNN) were used to carry out comparative analysis, alongside a cutting-edge, upcoming LSTM-CNN-based technique that shows innovative solutions. The inventive outcomes prove the effectiveness of these methodologies, reaching accuracy rates of 99.2% for CNN algorithm. Even though comparatively lower than CNN, the accuracy rates for LSTM-CNN and only LSTM approach were 97.6%, and 96.8%, respectively. The CNN-based approach is particularly endorsed for phishing detection.

The goal of this systematic review of Benavides et al., [6] was to provide readers, users, and other academics with an overview of a range of ideas made by earlier researches who have explored ways to combat such attacks through Deep Learning algorithms. This study makes two distinct contributions: it categorises safe solutions based on methodology of the paper, and it finds the prevalence of the URL-focused strategy. Deep neural networks and CNN appear as the most employed techniques.

In order to evaluate whether the provided URL is authentic or not, the study employs the supervised learning approach of machine learning.Ravindra et al., [17] made use of a dataset consisting of 2000 legitimate URLs and 2000 phishing URLs for their study.The Random Forest Algorithm is perfectly employed in this study along with a set of 9 characteristics for its efficient performance, resilience, and high accuracy. The system distinguishes whether the provided URL is a legitimate one or a phished one using distinctive classification.

To identify website phishing attacks, this paper suggests WebPhish, a complete DNN that is trained with implanted raw URLs and HTML text. In Opara et al., [13], the suggested model first automatically extracts the matching characters into homologous dense vectors using an embedding technique. The embedding matrices for HTML and URL are then combined by the concatenation layer. Convolutional layers are then employed to model the dependencies of its semantics. Extensive tests using actual phishing data produced an accuracy of 98.1%, proving that WebPhish performs better than standard detection methods for recognising phishing pages.

A novel approach of using K-Neighbours Classifier algorithm for the purpose of getting rid of outliers and making the technique robust to noisy data. It relies on the combinational hybrid approach of using K-Neighbours and SVM at the same time. The experimental results displayed in Altaher, A. et al., [3], showed an accuracy of 90.04\% for the hybrid approach using K-Neighbours and has proved quite effective for detection of phishing websites for both individuals and organisations. One of the prime reasons for using K-Neighbours in detection of phishing websites is its ability to handle large datasets which is generally the case for phishing website data. It is also able to capture complex and nonlinear patterns in the data which is crucial in analysing suspicious patterns.

Feroz et al., [7] aims to propose an approach that is automatically based on lexical and host-based features. The paper achieves an 93-97% accuracy by detecting many phishing hosts. One of the positive results includes the ability to keep a modest false positive rate. The paper examines the raw data obtained from the dataset and analyses the effectiveness of various feature subsets. The paper focuses on the chi-squared method and information gain attribute evaluation methods are used to increase the relevance of bigrams.

Mahajan et al.,[12] uses a fundamental way of machine learning for dubious website detection namely Decision Tree. It also uses Random Forest as it reduces the overfitting in decision trees and Support Vector Machine to overcome the outliers. The paper does a thorough competitive analysis of the machine learning algorithms by analysing the

algorithms based on accuracy of false positive and false negative rates. The paper successfully achieved an accuracy rate of 97.14% using the random forest algorithm with the lowest false positive rate. The paper's highlight is the usage of SHAP values to effectively understand the models applied to detect malicious aspects of a URL. The dataset in Puri et al.,[15] was fed into several classification models like K-means, CatBoost classifier, AdaBoost along with LightGBM classifier and others. CatBoost showed best results for accuracy and F1 value. The values of SHAP helped to determine the interpretation of the models used and to identify the crucial features in the model affecting the output.

The paper cohesively combines ML and DL algorithms to get a clean binary classification of the data and is an unprecedented method to approach the detection and classification of phishing websites. Actually, few have employed Long-Short Term Memory (LSTM), which detects phishing URLs using artificial neural networks. Research papers till date have not considered a targeted approach for URL phishing of the dataset that focuses on the Indian audience. With this paper, it is possible to determine malicious websites by incorporating different elements like handling varied data types, effective analysis of sequential URL data and capturing patterns using LSTM on Indian Phishing Website Dataset which have not been done before.

3. Methodology

This study employs a thorough methodology to investigate the realm of malicious websites. It begins with a thorough examination of existing literature and research to gain insight into the various forms and evolution of malicious websites. The methodology section begins by providing a description of the dataset. This is followed by a detailed explanation of the steps that were undertaken to clean the data, finally, the approach is implemented.

3.1. Dataset Description

The dataset used in the paper is extracted from Kaggle and is structured as 'URL' and 'Type' columns. It will be considered as primary dataset. The dataset has URLs which are categorized as phishing, benign, defacement and malware. However, the dataset used in the paper has been preprocessed to drop the 'defacement' and 'malware' types to obtain an effective binary classification between phishing and benign types of URL. Following the pre-processing procedure applied to the dataset, the phishing URLs are assigned the value of 1, while the benign URLs are given the value of 0. Next, the data undergoes normalization, which involves extracting the length of the URL data and organizing it in a column alongside the corresponding domain names. The approximate visual representation of the primary dataset is illustrated in Figure 1 .

This method introduced a secondary dataset which consists of around 80 data entries of malicious websites which fall under the category of phishing and the unique aspect of this is that all of them are of Indian origin.All these websites are from different domains like e-commerce, government document services, etc-.The aim of this paper is to create a dataset consisting of Indian malicious websites only as well as using them to predict other such websites.India is one of the major targets to such type of cyber attacks due to its population and the vulnerability of the people within, thus there is a need to safely detect and predict such calamities that occur in a country like India.

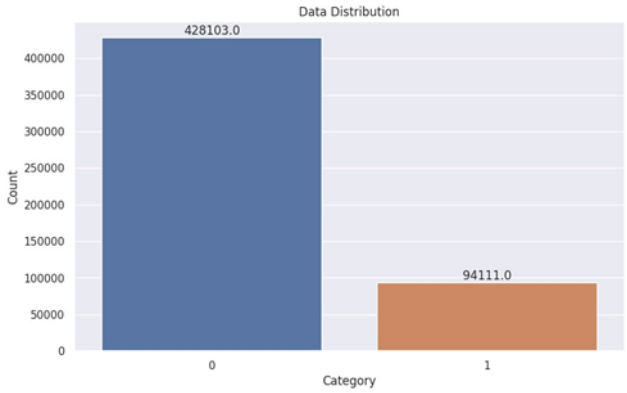

Fig .1. Data Distribution of Primary Dataset

3.2. Data Pre-Processing

The dataset used in the paper has been preprocessed to drop any label other than phishing and benign in the type column to get an efficient binary classification between the two. Further it was mapped from categorical to ordinal. The dataset classifies the phishing URL which being harmful as 1 and the non harmful or benign type with a value of 0.

3.3. Implementation

The URLs directly received cannot be used for Machine Learning Algorithms or Deep Learning algorithm. A degree of computation is needed. Thus, this research employs two methods, feature extraction and TF-IDF on the url links.

Feature extraction methods must be implemented on the malicious URL dataset where the special character symbols are extracted and counted. Then a column of abnormal URL is added onto the dataset that returns 1 if the URL name matches the hostname and 0 otherwise. After this, the data is sliced and categorised by the digits, letters, shortining service and IP address of the URL. The count is calculated and fed as a column input for the respective sections. After changes, the final dataset holds 522214 rows and 26 columns to be pre-processed. This is explained further in the Figure 2.

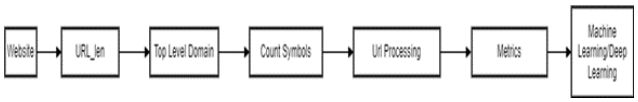

Fig. 2. Metric Formation for URL

URL Length (URL_len): This metric is used to find the total number of characters in a website's URL.

Top-Level Domain (TLD): Identify the top-level domain for instnace., .org, .edu ,.com of the website.

Symbol Counting: Count the occurrences of special symbols like '@', '?', '-', '=', '.', '\#', '\%', '+', '\$', '!', '*', ',', '//', '_', and ':' within the URL.

URL Processing: Analyze the URL for any suspicious or irregular patterns and structures, such as excessive symbols or unusual combinations, which may indicate potential malicious intent. Analyzing the digit count, letter count, Shortening Search like 'bit.ly,.net' et, having the IP address.

The study employs the TF-IDF (Term Frequency-Inverse Document Frequency) procedure for information retrieval from dataset. Term Frequency (TF) is defined as the frequency at which specific word or term occurs within a given document. Inverse Document Frequency (IDF) refers to the metric which evaluates the frequency of occurrence of terms within the dataset of malicious URLs. [16] The paper was inspired by the research study done on TF-IDF for detecting phishing websites. The paper uses TF-IDF as it is an effective method for text classification based on category by finding the topic. It addresses the limitations of the simple frequency counts and creates a distinctive categorization of the input data values.

Several research papers have used machine learning models to identify malicious URLs in various datasets. Nevertheless, the dataset used in the study encompasses a diverse range of machine learning models, including Decision Tree Classifier, Random Forest Classifier, SGD Classifier, Logistic Regression, and K-Neighbors Classifier. Additionally, the study incorporates advanced Deep Learning techniques such as Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) to forecast and evaluate the accuracy of the code.

3.3.1. Decision Tree Classifier

A decision tree classifier is a machine learning method used for classification tasks, which entail labelling or categorising an input following its attributes. Using a given criterion, the decision tree algorithm chooses the best feature to partition the data. The significance of each feature in the categorisation process can be figured out using decision trees. Because decision trees are so useful for feature selection and for figuring out which aspects affect an outcome most, the paper uses them to categorise the data as benign or phishing. [12] The paper takes notes from the research done on Decision Tree Classifier for detecting phishing URL.

3.3.2. Random Forest Classifier

In classification tasks, where the aim is to forecast a categorical outcome or label based on a set of input features, the random forest approach is most often utilised. [17] The excellent prediction accuracy, resilience, and versatility of the random forest to detect phishing URL in a study is one of the crucial reasons for opting for this machine learning algorithm. To choose features and determine which features are most pertinent to the classification assignment, the paper uses Random Forest, which offers a measure of feature relevance. It is resistant to outliers and noisy data since it integrates the predictions of several trees.

3.3.3. Stochastic Gradient Descent

Stochastic Gradient Descent Classifier, or SGD Classifier for short, is a machine learning technique used for classification tasks [5]. In binary classification issues, where it is desired to classify data points into one of many predetermined classes or categories, SGD is most frequently used. Since SGD does not require keeping the complete dataset in memory during training, it is appropriate for huge datasets. Due to the big size of the dataset used in the article, SGD is used to prevent overfitting.

3.3.4. Logistic Regression

When doing binary classification tasks, the objective of the classification process known as logistic regression is to estimate the likelihood that a given input data point belongs to one of two potential classes or categories [7]. Because logistic regression was created expressly for binary classification and effectively meets the criteria, it is used in the paper. Logistic regression can also handle huge data sets and data points as used in the paper.

3.3.5. K-neighbours

One of the simplest classification algorithms, the K-neighbours classifier makes predictions only based on the data. Because this classifier uses an instance-based learning technique, which can adapt to changes in the data without retraining the entire model [3], it is employed in the study.

3.3.6. Convolutional Neural Network

A special kind of artificial neural network called a convolutional neural network (CNN) is made for processing data that resembles grids. CNN has used methods like 1D convolutions applied to text embeddings to find applications in natural language processing, particularly in problems requiring text and sequential data. [6] In the research for detecting malicious URLs using Deep Learning methodologies, CNN is most often employed. Convolutional layer outputs are activated using ReLU (Rectified Linear Unit) activation functions to add non-

linearity to the network. CNNs can model intricate relationships in the data because of this non-linearity. CNN is able to create conclusive classifications and predictions thanks to its connected layers. For these reasons, the paper makes use of CNN for its advanced functionalities as compared to other Deep Learning techniques.

3.3.7. Recurrent Neural Network

An artificial neural network with the purpose of processing sequential data is called a recurrent neural network (RNN). The reason RNN is employed in this study is that it is very good at extracting significant information, such as patterns, character combinations, or certain character sequences that are suggestive of phishing URLs. Additionally, phishing URLs may hold character patterns or sequences that span several different places [10]. RNNs are suited for spotting such patterns that are indicative of phishing attempts because they are built to capture long-term dependencies.

3.3.8. Long short-term memory

The recurrent neural network (RNN) architecture known as LSTM, or Long Short-Term Memory, was created to handle sequential input and overcome some of the drawbacks of conventional RNNs. [2] The paper is inspired by a research study on use of LSTM for detecting phishing URL. Also, memory cells of LSTM enable them to store data for lengthy periods of time. This approach is used in the paper because LSTM networks are effective at handling sequential data. A relevant feature can be extracted from the data by treating each character or word in the URL as a time step in the sequence.

There are two datasets, introduced in the research, the primary dataset and the dataset created, the secondary Malicious Websites. The feature metrics are generated for both the dataset, followed by application of Machine Learning Algorithms and Deep learning Techniques. For easily formulating the pathway, to initially analyse the primary Dataset, 80% data for training on the models implemented and 20% as test data, for TF-IDF and Metrics. Then to compare the accuracy of training that is employed for the primary Dataset on the secondary Dataset, 100% training is done on primary and training purely on secondary Datasets, as described in the Figure 3.

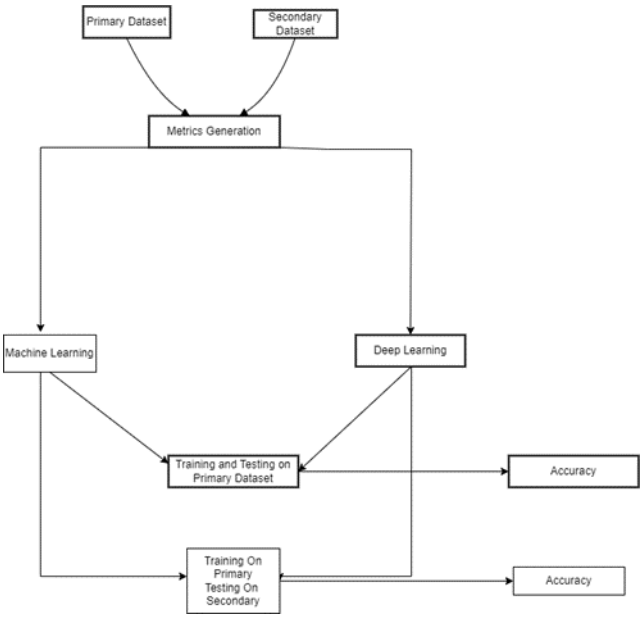

Fig. 3. Architecture

4. Results

The utilization of two distinct datasets was employed in the training of ML and DL models. The primary dataset was utilized for both the training and testing phases, employing an 80% portion for training and a 20% portion for testing.

4.1. ML

4.1.1. Primary

The machine learning algorithms, when trained and tested, presented the accuracies in the Table 1. According to the results of the methodology used in the creation of URL metrics, the most excellent accuracy was achieved in Random Forest Classifier, reaching an accuracy of 91.98%. Decision Tree Classifier achieved 91.45% accuracy while being slightly lower by around 1%. Logistic Regression and Stochastic Gradient Classifier, both showed a 6% decrease while keeping a similar accuracy of 85.06% & 84.98%, respectively. A 2% drop w.r.t. Random Forest was seen in the K-Nearest Neighbours Classifier, with 90.30% accuracy.

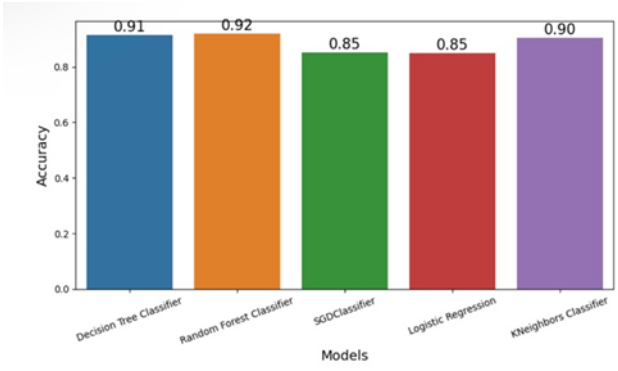

Fig. 4. Machine Learning Accuracies on Primary Dataset using Metric

Table 1. Model Comparison

Model	Accuracy(%)
Random Forest Classifier	91.98
Decision Tree Classifier	91.45
SGD Classifier	84.98
Logistic Regression	85.06
KNeighbours Classifier	90.30

After applying TF-IDF as a data preprocessing tool, the best result in this case was obtained for Random Forest Classifier with an accuracy 92.12%, followed by Decision Tree Classifier with an accuracy 91.45% which is comparable. A few more algorithms like K-Neighbours Classifier and Stochastic Gradient Descent Classifier 89.21% and 89.31% respectively are in the similar ranges, coupled by the employment of Logistic Regression model, which exhibits an accuracy rate of 90.82%. The graphical representation is as follows: -

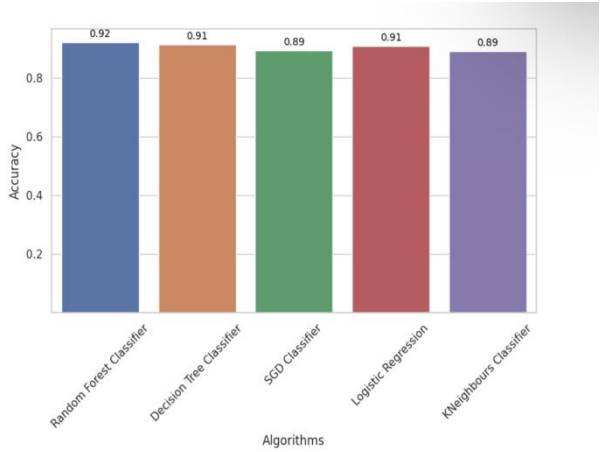

Fig. 5. Machine Learning Accuracies on Primary Dataset using TF-IDF

Table 2. Model Comparison

Model	Accuracy(%)
Random Forest Classifier	92.12
Decision Tree Classifier	91.45
SGD Classifier	89.31
Logistic Regression	90.82
KNeighbours Classifier	89.21

4.1.2. Secondary

The paper then delves into the methodology used in the creation of URL Metrics, training the algorithm on the primary dataset, and then testing it on the secondary dataset to obtain the results shown below. On applying various machine learning algorithms, we surprisingly observe that Decision Tree Classifier gives the highest accuracy of 83.33%, followed by Random Forest Classifier with 80.95%. Then a steady decline was seen, with K-Neighbours Classifier giving 72.62% accuracy while Stochastic Gradient Classifier and Logistic Regression give 45.24% accuracy. From this, it can be inferred that there is a deficiency of secondary dataset in the training set. The graphical representation is as follows:

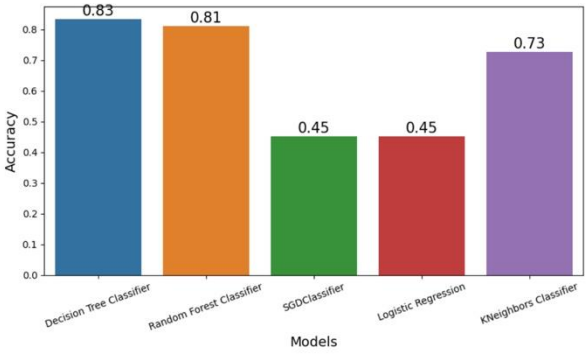

Fig. 6. Machine Learning Accuracies on Secondary Dataset using Metric

Table 3. Model Comparison

Model	Accuracy(%)
Random Forest Classifier	80.95
Decision Tree Classifier	83.33
SGD Classifier	45.24
Logistic Regression	45.24
KNeighbours Classifier	72.62

Following the use of TF-IDF for data preprocessing, the best results were obtained for Random Forest Classifier and Decision Tree Classifier with a precision of 82.33%and 82.4%, respectively. Then, for K-Neighbours Classifier whose precision was 64.29%, a substantial decrease. A few more algorithms, such as Logistic Regression and Stochastic Gradient Descent Classifier, achieved accuracy of 48.81% and 46.53%, respectively, representing a further decrease in accuracy. The diagram is as follows: -

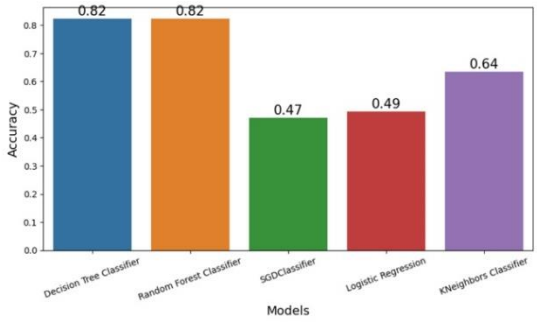

Fig. 7. Machine Learning Accuracies on Secondary Dataset using TF-IDF

Table 4. Model Comparison

Model	Accuracy(%)
Random Forest Classifier	82.33
Decision Tree Classifier	82.14
SGD Classifier	46.53
Logistic Regression	48.81
KNeighbours Classifier	64.29

4.2. DL

4.2.1. Primary

The study makes use of the primary dataset, with an 80:20 split of training and testing data. The training process makes use of deep learning algorithms such as Convolutional Neural Network, Recurrent Neural Network, and Long-Short Term Memory.Using effective feature extraction methods like symbol counting, top-level domain, hostname, etc the dataset is pre-processed before training. The paper uses relu activation function with optimiser as adam, loss is calculated using categorical cross-entropy. The models run for 10 epochs each. Out of the three deep learning models, LSTM observes the highest accuracy of 90.99%. The accuracies shown by CNN and RNN vary slightly with 86.91% and 86.49% respectively. The graphical representation is as follows: -

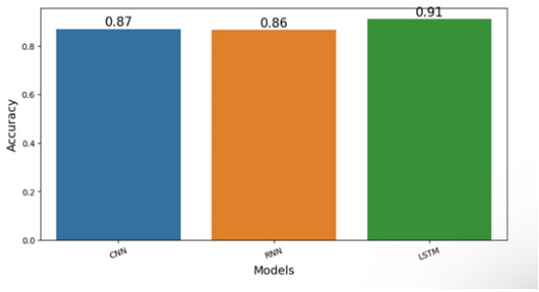

Fig. 8. Deep Learning Accuracies on Primary Dataset

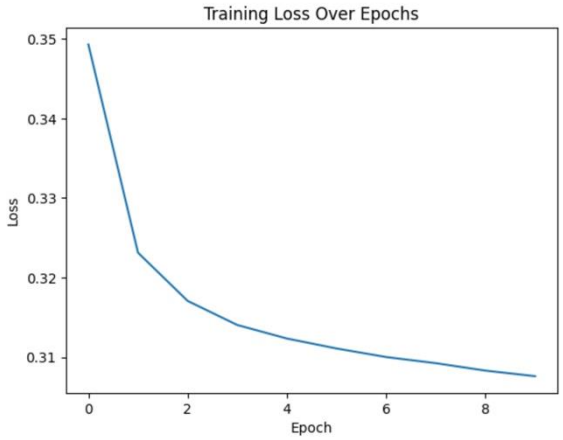

Fig. 9. Epoch Loss Curve for CNN

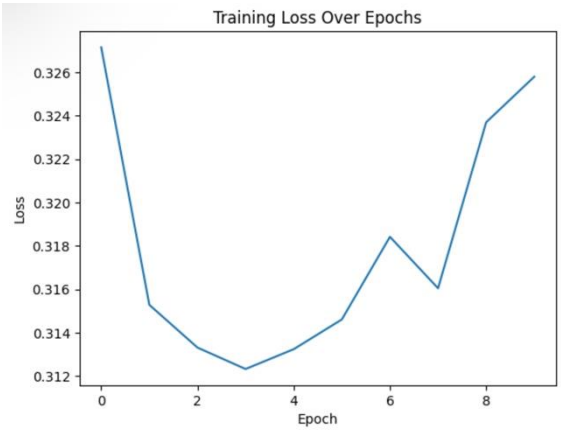

Fig. 10. Epoch Loss Curve for RNN

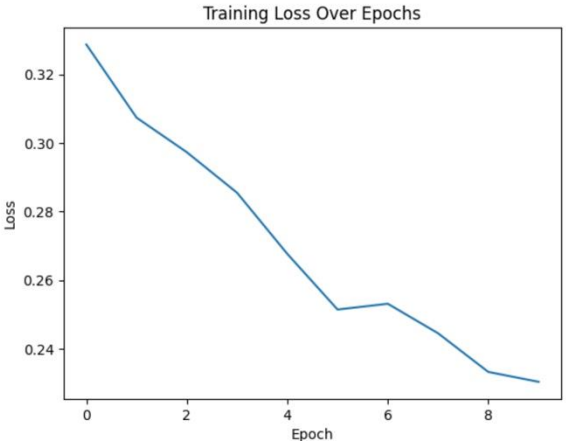

Fig. 11. Epoch Loss Curve for LSTM

Table 5. Model Comparison

Model	Accuracy(%)
CNN	86.91
RNN	86.49
LSTM	90.99

4.2.2. Secondary

The paper trains on the primary dataset before testing the curated secondary dataset with deep learning algorithms such as Convolutional Neural Network, Recurrent Neural Network, and Long-Short Term Memory. The dataset is pre-processed before training using effective feature extraction methods such as symbol counting, top-level domain, hostname, and so on. The paper employs the relu activation function with adam as the optimizer, and the loss is computed using categorical crossentropy. Each model runs for ten epochs. Out of the three deep learning models, LSTM observes the highest accuracy of 80.95%. The accuracies proved by CNN and RNN differ significantly, with 47.61% and 45.23%, respectively. This implies that there is a scarcity of secondary datasets in the training set. The graphical representation is as follows: -

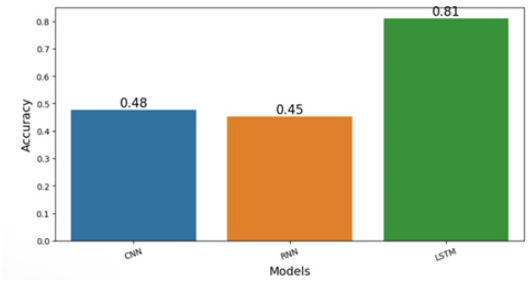

Fig. 12. Deep Learning Accuracies on Secondary Dataset

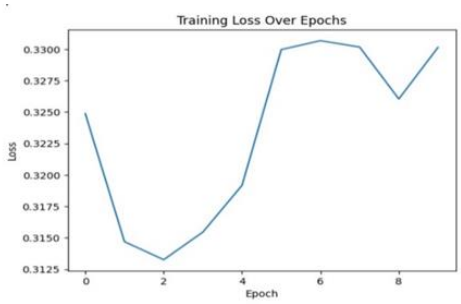

Fig. 13. Epoch Loss Curve for CNN

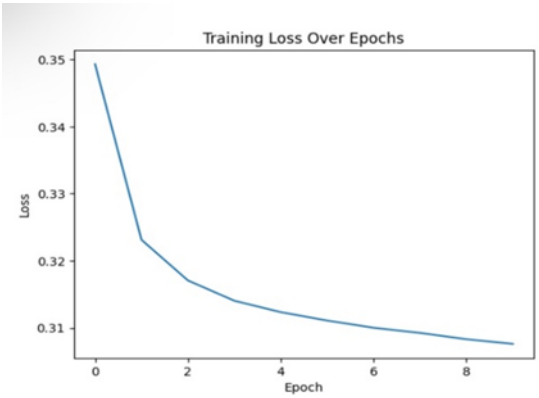

Fig. 14. Epoch Loss Curve for RNN

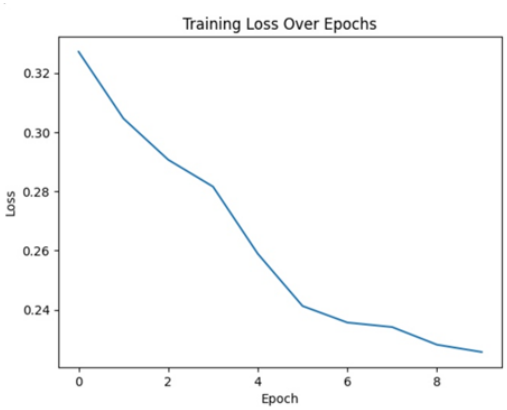

Fig. 15. Epoch Loss Curve for LSTM

Table 6. Model Comparison

Model	Accuracy(%)
CNN	47.61
RNN	45.23
LSTM	80.95

5. Conclusion

This work emphasizes the importance of regionally focused cybersecurity research and dataset creation. Not only are defenses against malicious websites strengthened by addressing these regional nuances, but it also contributes to the development of context-aware solutions capable of safeguarding online experiences in diverse global contexts. As the digital landscape evolves, the dedication to improving web security remains, ensuring a safer and more resilient online environment for all. In summary, the creation of an open-source dataset for malicious websites in India represents a noteworthy advancement in the realm of cybersecurity. By providing public access to this dataset, the initiative facilitates collaborative endeavors aimed at enhancing web security and establishes the foundation for its continuous growth and enhancement.The study revealed an important finding: traditional methodologies and models that have proven effective with more generalized datasets, such as those available on platforms like Kaggle, frequently encounter limitations when applied to Indian datasets. The unique characteristics and complexities of the Indian web environment necessitate customized approaches and regional insights. This highlights the importance of datasets tailored to specific regions in addressing India's unique challenges posed by web threats. Therefore, the paper initiates a small step in a big journey to provide its contribution to completely eradicate phishing from a major hub like India.

Author contributions

All authors have contributed equally to this research .

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] Aljofey A., Jiang Q., Qu Q., Huang M., \& Niyigena J. P. 2020. An effective phishing detection model based on character level convolutional neural network from URL. {Electronics}, 9(9), 1514.

[2] Alshingiti Z., Alaqel R., Al-Muhtadi J., Haq Q.E.U., Saleem K., Faheem M.H. 2022. A Deep Learning-Based Phishing Detection System Using CNN, LSTM, and LSTM-CNN. {Electronics} 2023, 12, 232.

[3] Altaher A. 2017. Phishing websites classification using hybrid SVM and KNN approach.{ International Journal of Advanced Computer Science and Applications}, 8(6)..

[4] Arathi Krishna V, Anusree A, Blessy Jose, Karthika Anilkumar, Ojus Thomas Lee, 2021, Phishing Detection using Machine Learning based URL Analysis: A Survey, {INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) NCREIS – 2021} (Volume 09 – Issue 13),

[5] Babu Rao Pawar, Nagasunder Rao Pawar. Detection of Phishing URL using Machine Learning. Diss. Dublin, National College of Ireland, 2021.

[6] Benavides E., Fuertes W., Sanchez S., Sanchez M. 2020. Classification of Phishing Attack Solutions by Employing Deep Learning Techniques: A Systematic Literature Review. In: Rocha, Á., Pereira, R. (eds) Developments and Advances in Defense and Security. Smart Innovation, Systems and Technologies, vol 152. Springer, Singapore.

[7] Feroz M. N., & Mengel S. 2014, October. Examination of data, rule generation and detection of phishing URLs using online logistic regression. In 2014 {IEEE International Conference on Big Data (Big Data)} (pp. 241-250). IEEE.

[8] Jain A. K., & Gupta B. B. 2018. PHISH-SAFE: URL Features-Based Phishing Detection System Using Machine Learning.{Cyber Security}, 467–474. doi:10.1007/978-981-10-8536-9_44

[9] Jeeva S.C., Rajsingh E.B. Intelligent phishing url detection using association rule mining. {Hum. Cent. Comput. Inf. Sci.} 6, 10 2016. https://doi.org/10.1186/s13673-016-0064-3

[10] Kalaharsha P. and Babu M. Mehtre. "Detecting Phishing Sites--An Overview." {arXiv} preprint arXiv:2103.12739 2021.

[11] M. N. Feroz and S. Mengel, "Phishing URL Detection Using URL Ranking," 2015 {IEEE International Congress on Big Data}, New York, NY, USA, 2015, pp. 635-638, doi: 10.1109/BigDataCongress.2015.97.

[12] Mahajan, Rishikesh & Siddavatam, Irfan. 2018. Phishing Website Detection using Machine Learning Algorithms. {International Journal of Computer Applications.} 181. 45-47. 10.5120/ijca2018918026.

[13] Opara, Chidimma & Chen, Yingke & wei, Bo. 2020. Look Before You Leap: Detecting Phishing Web Pages by Exploiting Raw URL And HTML Characteristics.

[14] Orunsolu A. A., Sodiya A. S., & Akinwale A. T. 2022. A predictive model for phishing detection. {Journal of King Saud University-Computer and Information Sciences}, 34(2), 232-247.

[15] Puri N., Saggar P., Kaur A., & Garg P. 2022, July. Application of ensemble Machine Learning models for phishing detection on web networks. In {2022 Fifth International Conference on Computational Intelligence and Communication Technologies (CCICT)} (pp. 296-303). IEEE.

[16] Rao, Routhu Srinivasa, Tatti Vaishnavi, and Alwyn Roshan Pais. "CatchPhish: detection of phishing websites by inspecting URLs." {Journal of Ambient Intelligence and Humanized Computing} 11 2020: 813-825.

[17] Ravindra, Salvi & Sanjay, Shah & Gulzar, Shaikh & Pallavi, Khodke. 2021. Phishing Website Detection Based on URL. {International Journal of Scientific Research in Computer Science, Engineering and Information Technology}. 589-594. 10.32628/CSEIT2173124.

[18] Sahingoz O. K., Buber E., Demir O., & Diri B. 2019. Machine learning based phishing detection from URLs. {Expert Systems with Applications}, 117, 345-357.

[19] Sánchez-Paniagua M., Fernández E. F., Alegre E., Al-Nabki W., & Gonzalez-Castro V. 2022. Phishing URL detection: A real-case scenario through login URLs. {IEEE Access}, 10, 42949-42960.

[20] S. Patil and S. Dhage, "A Methodical Overview on Phishing Detection along with an Organized Way to Construct an Anti-Phishing Framework," {2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS)}, Coimbatore, India, 2019, pp. 588-593, doi:

10.1109/ICACCS.2019.8728356.

[21] Tang L., & Mahmoud Q. H. 2021. A survey of machine learning-based solutions for phishing website detection. {Machine Learning and Knowledge Extraction}, 3(3), 672-694.

[22] Ubing A. A., Jasmi S. K. B., Abdullah A., Jhanjhi N. Z., & Supramaniam M. 2019. Phishing website detection: An improved accuracy through feature selection and ensemble learning.

[23] Vinodharan, Chandanaboina. 2023. Phishing attack and Phishing Url Detection. 10.13140/RG.2.2.19036.46728.

[24] Wei W., Ke Q., Nowak J., Korytkowski M., Scherer R., & Woźniak M. 2020. Accurate and fast URL phishing detector: a convolutional neural network approach. {Computer Networks}, 178, 107275.