WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

DESIGN AND DEVELOP AN ENSEMBLE LEARNING SCHEME TO DETECT THE ANOMALY AND MALICIOUS URL USING DEEP LEARNING

1st Shivakumar, 4th Ambarish K C , 2nd Monith L , 5thSwetha T , 3rd Kishor T M

[1] (Student) *Department of computer science and Engineering* S J C Institute of Technology Chikkaballapur, India shiva98457022@gmail.com
[4] (Student) *Department of computer science and Engineering* S J C Institute of Technology Chikkaballapur, India chandurajchandu06@gmail.com
[2] (Student) *Department of computer science and Engineering* S J C Institute of Technology Chikkaballapur, India monithmanu1344@gmail.com
[5] (Scholar) *Department of computer science and Engineering* S J C Institute of Technology Chikkaballapur, India swetha102reddy@gmail.com
[3] (Student) *Department of computer science and Engineering* S J C Institute of Technology Chikkaballapur, India kishortm17@gmail.com

ABSTRACT—

Malicious Uniform Resource Locator (URL), also known as a malicious website, serves as a primary platform to host unwanted content like spam, malicious ads, phishing, and drive-by exploits, among others. Detecting these malicious URLs promptly is crucial. While past research has utilized blacklisting, regular expressions, and signature matching methods, these approaches prove ineffective in identifying variations of existing malicious URLs or entirely new ones. A solution based on machine learning is vital to address this issue.

This particular solution requires thorough research into feature engineering and representation of security artifact types, such as URLs. Resources for feature engineering and representation need constant updating to handle new and evolving URLs efficiently. In recent times, deep learning has enabled artificial intelligence (AI) systems to match and even surpass human-level performance in various domains, including computer vision applications.

To harness the potential of AI for cybersecurity, we introduce Deep URL Detect (DUD), a method that encodes raw URLs using character-level embedding. Such embedding, a cutting-edge technique in natural language processing (NLP), represents characters in a numerical format. Deep learning architectures leverage hidden layers to extract features from character-level embeddings. Subsequently, a feedforward network with a non-linear activation function estimates the likelihood of a URL being malicious.

Our work evaluates multiple state-of-the-art deep learning-based character-level embedding techniques for detecting malicious URLs. By conducting various experiments, we have selected the most optimal deep learning-based character-level embedding model. The performance of our proposed method, DUD, proves to be comparable and computationally efficient compared to other deep learning-based character-level embedding approaches in all scenarios. Furthermore, architectures based on character-level embedding models outperform n-gram representation due to their ability to capture the sequence and relationships among all URL characters.

Keywords—Cybercrime, Cyber security, Deep learning, Malicious URL, Machine learning, Character embedding.

Introduction :

MALICIOUS Uniform Resource Locator (URL) host hosts unsolicited information and attackers utilize malicious URLs as a primary tool in perpetrating cyber-crimes. Email and social media platforms like Facebook, Twitter, WhatsApp, Orkut etc. are the most popular applications for disseminating these malicious URLs. These platforms host unsolicited information on their pages. When an unwary user inadvertently visits the website through the URL, their system will be compromised, rendering them susceptible to various forms of fraud such as malware installation, data and identity theft. Malicious URLs have resulted in billions of dollars' worth of losses annually. Hence, the need for effective detection techniques to promptly identify malicious URLs and alert the network administrator.

Most of the available commercial products in the market rely on blacklisting. This method involves a database that stores a list of known malicious URLs. The list is constantly updated by antivirus groups through scanning and crowdsourcing methods. While this approach accurately detects existing malicious URLs stored in the database, it fails to identify new variants or entirely new malicious URLs. Presently, malicious actors employ mutation strategies to generate multiple versions of existing malware. To address this challenge, machine learning methodologies are employed. Currently, the predominant method involves leveraging domain expertise to extract lexical URL features, followed by the application of machine

learning algorithms. The most commonly utilized feature engineering technique is Bag-of-Words (BoW) and the prominent machine learning model is Support Vector Machine (SVM). Despite being an alternative to blacklisting, machine learning-based solutions encounter several challenges:

1. Conventional URL representation techniques struggle to capture sequential patterns and relationships among characters.
2. Traditional machine learning models depend on manual feature engineering, demanding extensive cyber security domain knowledge and being deemed a daunting task. Feature-based approaches are insecure in hostile environments.
3. These solutions lack the capacity to recognize latent features and may fail to generalize on new test data. Moreover, due to the significantly large number of unique words, memory limitations arise during training of the machine learning model.

LITERATURE SURVEY

Our contributions and engagements in this work are as follows: The proposed model in this study introduces a DCNN-based approach for detecting harmful URLs. Through the utilization of the dynamic convolution method, a unique folding layer is introduced to the complex multilayer convolution structure. The traditional pooling layer is replaced with a k-max-pooling layer in this novel model. The breadth of the vector entrance dimension impacts the core layer of the dynamic convolution process significantly. To ensure a more comprehensive analysis across a wide range, adjustments in the pooling layer values are influenced by the depth of the current convolution layer and the size of the input URL. During the process of feature extraction and representation, components are obtained from the sequence of URLs. The convolutional neural network swiftly processes and integrates variables to form a vector for model assessment. This methodology integrates both personality embedding and word embedding strengths while streamlining the feature extraction process and eliminating the need for manual feature extraction. Phrase sequences are collected through word embedding, while character embedding is capable of handling unfamiliar characters and words in URLs. By reducing the dictionary size and vector dimensions, the model can more precisely identify URLs and allocate memory space for better information extraction. Numerous comparative experiments were conducted to validate the efficacy of the proposed model. The findings suggest that utilizing personality-based embedding achieves higher accuracy levels than combining phrase and personality embedding. Detailed testing procedures were carried out, displaying the enhanced impact of employing a CNN-based architecture and specific URL parameters in the model.

Approaches for Detecting Malicious URLs

Two main categories of methods exist for identifying harmful URLs: traditional techniques that solely depend on blacklisting and methods that heavily lean on machine learning.

The primary method of detecting listed sites is initially discussed in the literature. Despite its simplicity and effectiveness, this approach is limited and fails to detect newly generated harmful URLs.

Current literature continues to highlight the existence of two major techniques for recognizing malicious URLs: standard approaches that exclusively rely on blacklisting and those that extensively use machine learning.

While the original method of detecting listed URLs is first mentioned in literature, its effectiveness is constrained and fails to identify recently created harmful URLs.

METHODOLOGY

MACHINE LEARNING & DEEP LEARNING

Due to it's ability to handle enormous quantities of data, machine learning & deep learning have become increasing powerful tools in recent years.

SOFTWARE REQUIREMENTS:

- Accurate classification: The system should be able to classify URLs into Benign and Malicious accurate.
- Accessibility: The User should be able to access application to enter URLs for classification.
- User friendly: The software should be user-friendly for users who are not well-versed in computer science.

CNNs and RNNs have been included into neural network structures to fulfill this purpose. The diagram appears as follows: RNNs and LSTMs are examples of sequence generator architectures that can begin by translating a feature vector with a set length from either a picture. To prepare a list of labels or terms for your image, follow this procedure. The encoder used for this project is ResNet50. Using pre-trained algorithms, the millions of images in the ImageNet dataset were separated into 1000 teams. Replace the top layer, which contains 1000 neurons and is only used for ImageNet classification, with a linear layer containing a double the neurons. as add to use this network. This is because the network's weights are adjusted to identify numerous features common to nature. Several neurons create through LSTM. Long Short-Term Memory (LSTM) cells compose an RNN, which is used to generate captions continuously from raw images. To remember things from earlier steps, these cells employ the repetition and gate ideas. For some further detail, you may read or watch this. The outputs the output layer, typically predicts the subsequent word based based on the visuals and present flow, obtains the combined output from the encoder and decoder!

GENERATING THE MODELS: The dataset makes it clear that this is a supervised machine learning and deep learning task. Two of the most prevalent kinds of supervised learning problems are classification and regression. • There is a classification problem for the data collection method that was employed because the input URL is categorized as either malicious (1) or real (2). (0). When training the dataset, the following categories of

supervised learning models (classifications) were considered: XGBoost, Random Forest, Decision Trees, Multilayer Perceptrons, and Support Vector Machines

PROJECT DESIGN/MODEL:

Importing the Dataset: 6000 of the 11000 URLs in the experimental dataset for the malicious URL identification model have been identified as malicious, and the remaining 5000 as secure. Each of these URLs was examined by the Virus Total tool in order to confirm the markings present on each URL. The entire dataset has been preserved in CSV format. URLhaus, Phishtank, Alexa, and Malicious_n_Non-Malicious URL are a few of the data sources. • Model training: The dataset is divided into two subsets: secure and harmful URLs. Eighty percent of the data is used for training, while twenty percent is used for testing. The trial is run several times for every algorithm. • Producing the Forecasts: Test data is used to evaluate the models following the training phase.

All of the algorithms are tested, and the proportion of correctly predicted values as well as the difference between the predictions and actual values are used to measure the algorithms' performance. • Assessing the model and drawing comparisons: For each algorithm, the performance in testing and training is assessed. The algorithms' accuracy is gauged using the accuracy score.

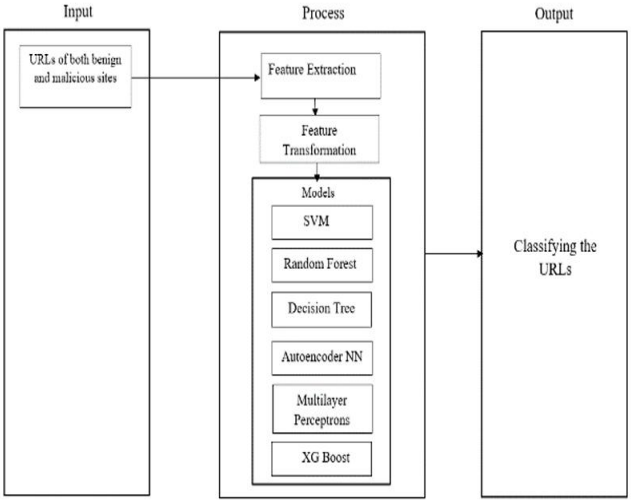

Figure (1): Workflow Diagram

The accuracy rate is the proportion of right choices made across all test sets.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

where:

- TP- The number of malicious-URLs that were accurately labelled is a true positive.
- FN - The quantity of malicious-URLs mistakenly labelled as secure is known as false negative.
- TN- True negatives is the quantity of properly labelled safe URLs.
- FP - False positives are instances where secure URLs are mistakenly labelled as malicious.

Following evaluation, each model's performance is evaluated, and the top-performing model is selected for web framework implementation.

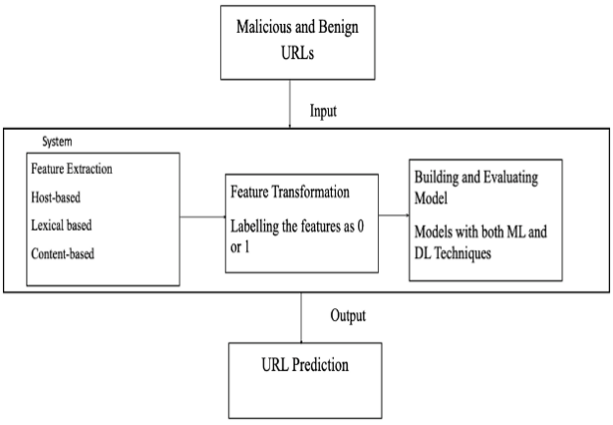

Figure (2): Component Design

SOFTWARE IMPLEMENTATION:

Step 1: Importing the Dataset Step

Step 2: Extraction of Features

Step 3: Converting Features

Step 4: Model Creation

Step 5: Model Training Step

Step 6: Prediction Generation

Step 7: Assessing the model and making comparisons.

RESULTS AND DISCUSSION

A model's accuracy value determines its rating. In comparison to all input samples, it displays the percentage of accurate forecasts that were created. On the basis of training and assessment accuracy, compare every model. An accurate method for identifying and categorizing benign and malicious URLs uses machine learning and deep learning.

ML Model	Train Accuracy	Test Accuracy
Multilayer Perceptrons	0.875	0.865
XGBoost	0.876	0.863
Decision Tree	0.825	0.814
Random Forest	0.823	0.810
SVM	0.787	0.777

Figure (3): Accuracy of the Models

The Voting methodology's accuracy and dependability make it feasible to distinguish between harmful and secure URLs. All models' AUCs function with very little variation. This could make it easier to find dangerous URLs more frequently. This still stands as one of the primary avenues for further expansion. They were all text representations that performed better than the DNN-based three-gram approach and made use of character level Keras encoding. Compared to traditional machine learning-based techniques that depend on manually created features, deep learning in conjunction with hacking-based harmful URL identification can be a dependable substitute. This could make it easier to find dangerous URLs more frequently. This still stands as one of the primary avenues for further expansion. They were all text representations that performed better than the DNN-based three-gram approach and made use of character level Keras encoding. Compared to traditional machine learning-based techniques that depend on manually created features, deep learning in conjunction with hacking-based harmful URL identification can be a dependable substitute. This is a result of the ability of malevolent writers to understand the features that were manually generated in an attempt to evade detection by using topic expertise.

CONCLUSION

Using rogue websites that mimic trustworthy URLs and online pages is a common social engineering technique. The objective of this project is to train deep neural networks and machine learning models using the available data sets to anticipate hazardous websites. The website's content-based functionality and the targeted URLs are gathered from a dataset that includes the website's benign and hazardous URLs. Determine the performance level of each model and draw comparisons. This research combines machine learning and deep learning approaches to more accurately predict harmful URLs.

REFERENCES :

[1] R. B. Basnet, D. J. Lemay and T. Doleck, “Examining the relationship between threat and coping appraisal in phishing detection among college students,” Journal of Internet Services and Information Security, 2020.

[2] H. Kim, “5G core network security issues and attack classification from a network protocol perspective,” Journal of Internet Services and Information Security, 2020.

[3] R. Shash and R. B. Basnet, “Towards detecting and classifying network intrusion traffic using deep learning frameworks,” Journal of Internet Services and Information Security, 2019.

[4] J. O. SoK and K. Aram, “A systematic review of insider threat detection,” Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA), 2019.

[5] M. Cheminod and F. Valenza, “An optimized firewall anomaly resolution,” Journal of Internet Services and Information Security 2020.

[6] G. Warner, L. F. Cranor, J. Hong, S. Sheng, B. Wardman and C. Zhang, “An empirical analysis of phishing blacklists,” in Proceedings of Sixth Conference on Email and Anti-Spam (CEAS), 2009.

[7] P. Komisarczuk, C. Seifert and I. Welch, “Identification of malicious web pages with static heuristics,” in Conference on Telecom Networks and Applications, 2008. ATNAC 2008. Australasian. IEEE, 2008.