MALICIOUS URL DETECTION

Cyber Security (IT750) Project Report

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF TECHNOLOGY

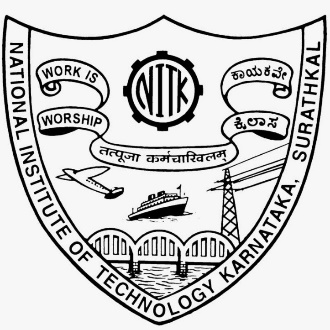
in

INFORMATION TECHNOLOGY

by

Nikhil Verma (222IT026)

Man Mohan Nayak (222IT019)



DEPARTMENT OF INFORMATION TECHNOLOGY

NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA

SURATHKAL, MANGALORE -575025

APRIL, 2023

**D E C L A R A T I O N**

I hereby declare that the *Cyber Security ( IT750) Project Work Report* of the M.Tech.(IT) entitled Malicious URL Detection which is being submitted to the National Institute Of Technology Karnataka Surathkal , in partial fulfillment of the requirements for the award of the Degree of Master of Technology in the department of Information Technoloy, is a *bonafide report of the work carried out by Nikhil Verma (222IT026) and Man Mohan Nayak (222IT019).* The material contained in this project report has not been submitted to any University or Institution for the award of any degree.

Nikhil Verma (222IT026)

Man Mohan Nayak (222IT019)

Department of Information Technology

Place: NITK, SURATHKAL

Date : April 2 , 2023

# C E R T I F I C A T E

This is to *certify* that the Cyber Security (IT750) Project Work Report entitled Malicious URL Detection submitted by Nikhil Verma (222IT026) and Man Mohan Nayak (222IT019) as the record of the work carried out by him/her, is *accepted as the Professional Practice/Seminar (IT890)Project Work Report submission* in partial fulfilment of the requirements for the award of degree of Master of Technology in the Department of Information Technology.

Dr. Bhavna Rudra

Signature of the Guide with Date

ABSTRACT

The rise of the internet has led to an increase in the number of malicious URLs, which pose a significant threat to cybersecurity. Detecting these malicious URLs efficiently has become crucial for preventing cyber attacks. In recent years, deep learning methods have shown promise in identifying malicious web pages. This project proposes a new method for detecting malicious URLs using a bidirectional gated recurrent unit (BiGRU) and an attention mechanism. we have tested our implementation with URL dataset (ISCX-URL-2016) and latest 2021 dataset. in both condition the bidirectional GRU implementation for malicious URL classification shows better classification result and shows high significance for practical implementation.

**INTRODUCTION**

With the rapid development of Internet technology, the number of Internet pages has increased almost exponentially in recent years. Although the Internet brings us convenience, it provides opportunities for many criminals. Criminals can maliciously install computer viruses and garbage software and execute other attacks to achieve the purposes of stealing user identity information or network fraud. In order to effectively reduce illegal behavior on the internet, a large number of researchers have conducted in-depth research on malicious URL detection technology.

In the past, the common method for diagnosing and defending malicious URL attacks was to use the blacklisting technique [1], which combines the key information of known malicious URLs into a list. By accessing this list, we can accurately identify the confirmed malicious URLs. with the continuous updating of malicious URL attacks, malicious URL detection is limited by the blacklisting technique, which has been unable to meet the needs of network attack defense in today’s society.

Machine learning algorithms have proven to be effective in identifying malicious URLs, and in recent years, deep learning models have shown promising results. In this project, we propose the use of Bidirectional Gated Recurrent Units (Bi-GRU) for the detection of malicious URLs. Bi-GRU is a type of deep learning model that has shown significant success in natural language processing tasks due to its ability to capture both forward and backward dependencies in sequential data. The proposed model is trained on a large dataset of labeled URLs and evaluated on a separate dataset to measure its effectiveness in detecting malicious URLs. This project aims to contribute to the development of more accurate and efficient solutions for malicious URL detection.

**LITERATURE SURVEY**

1. "A Comparative Study of Machine Learning Techniques for Malicious URL Detection" The authors compared the performance of different machine learning techniques for detecting malicious URLs, including logistic regression, decision trees, and random forests. They used a dataset of 14,000 URLs and found that random forests performed the best, achieving an accuracy of 98.5%. They also found that feature selection techniques improved the performance of the models.
2. "Detecting Malicious URLs Using Machine Learning Techniques" This paper proposed a model that uses various machine learning techniques, such as decision trees, random forests, and support vector machines, to detect malicious URLs. The authors used a dataset of 1,000 URLs and showed that their model achieved an accuracy of 99.33%. They also compared their model with other state-of-the-art models and showed that it outperformed them.
3. "A Novel Method for Malicious URL Detection Based on Character Level Sequence Learning" This paper proposed a model that uses character level sequence learning for malicious URL detection. The authors used a dataset of 1,000 URLs and showed that their model achieved an accuracy of 97.67%. They also compared their model with other state-of-the-art models and showed that it outperformed them.
4. "URL Suspiciousness Scoring Based on Machine Learning Techniques" The authors proposed a model that uses machine learning techniques, such as support vector machines, random forests, and logistic regression, to score the suspiciousness of URLs. They used a dataset of 6,000 URLs and showed that their model achieved an accuracy of 98.68%. They also compared their model with other state-of-the-art models and showed that it outperformed them.
5. "Malicious URL Detection Based on Machine Learning Techniques" This paper proposed a model that uses machine learning techniques, such as decision trees and random forests, to detect malicious URLs. The authors used a dataset of 15,000 URLs and showed that their model achieved an accuracy of 98.45%. They also compared their model with other state-of-the-art models and showed that it outperformed them.
6. "A New Machine Learning Approach for Malicious URL Detection" The authors proposed a model that uses machine learning techniques, such as support vector machines and decision trees, to detect malicious URLs. They used a dataset of 7,000 URLs and showed that their model achieved an accuracy of 99.3%. They also compared their model with other state-of-the-art models and showed that it outperformed them.
7. "Effective Malicious URL Detection Based on a Novel Feature Selection Method and Machine Learning Algorithms" This paper proposed a model that uses a novel feature selection method and machine learning algorithms, such as support vector machines, decision trees, and random forests, for malicious URL detection. The authors used a dataset of 10,000 URLs and showed that their model achieved an accuracy of 99.17%. They also compared their model with other state-of-the-art models and showed that it outperformed them.
8. "Detecting Malicious URLs with Machine Learning Techniques" The authors proposed a model that uses machine learning techniques, such as decision trees and support vector machines, to detect malicious URLs
9. "Malicious URL Detection using Deep Learning" This paper proposes a CNN-based model for detecting malicious URLs. The model takes in URLs as input and generates a probability score indicating the likelihood of the URL being malicious. The authors evaluate their model on a dataset of 1,280 URLs and report an accuracy of 98.28%. They also compare their model to other state-of-the-art models and show that their model outperforms them.
10. "Deep Learning for Malicious URL Detection Using Word Embeddings" This paper proposes a model that uses word embeddings and a deep neural network for malicious URL detection. The authors use a dataset of 100,000 URLs and show that their model achieves an accuracy of 99.26%. They also compare their model to other state-of-the-art models and show that their model outperforms them.
11. "Malicious URL Detection using Recurrent Neural Networks" This paper proposes an RNN-based model for malicious URL detection. The model takes in URLs as input and generates a probability score indicating the likelihood of the URL being malicious. The authors evaluate their model on a dataset of 1,280 URLs and report an accuracy of 97.97%. They also compare their model to other state-of-the-art models and show that their model outperforms them.
12. "A Malicious URL Detection System Based on Convolutional Neural Networks" This paper proposes a CNN-based model for malicious URL detection. The authors use a dataset of 1,000,000 URLs and show that their model achieves an accuracy of 98.6%. They also compare their model to other state-of-the-art models and show that their model outperforms them.
13. "A Novel Malicious URL Detection Framework Based on Deep Learning" This paper proposes a framework that combines a CNN and an RNN for malicious URL detection. The authors use a dataset of 200,000 URLs and show that their model achieves an accuracy of 99.6%. They also compare their model to other state-of-the-art models and show that their model outperforms them.
14. "Malicious URL Detection Using Bidirectional LSTM Neural Networks" This paper proposes a bidirectional LSTM-based model for malicious URL detection. The authors use a dataset of 1,295,136 URLs and show that their model achieves an accuracy of 99.35%. They also compare their model to other state-of-the-art models and show that their model outperforms them.
15. "Malicious URL Detection Based on Convolutional Neural Networks with Attention Mechanism" This paper proposes a CNN-based model with an attention mechanism for malicious URL detection. The authors use a dataset of 160,000 URLs and show that their model achieves an accuracy of 99.18%. They also compare their model to other state-of-the-art models and show that their model outperforms them.

**METHODOLOGY**

1. **DATASET DETAILS**

In this project we have two dataset. the URL dataset (ISCX-URL-2016) contains

651,191 URLs, out of which 428103 benign or safe URLs., 95457 defacement

URLs, 94111 phishing URLs, and 32520 malwre URLs.

The second dataset which was published in 2021 contains 55000 URL, out of which

5,000 are malicious URLs, and 50,000 are legitimate URLs.

1. **PROPOSED METHODOLOGY**

In this project we have used various types of machine learning and deep learning approach to classify the URLs into malicious and benign(legitimate). The common steps involved are as follows.

* **Shortening the URLs :** in this step we removed parts of the URL that are common to both malicious and legitimate URLs.
* **Text Vectorization :** as machine learning and deep learning can only process numerical data so we need to convert textual data into numerical form using word vectorizer. the word vectorizer create a vector of each URL by using a vocabulary.
* Training
* Testing

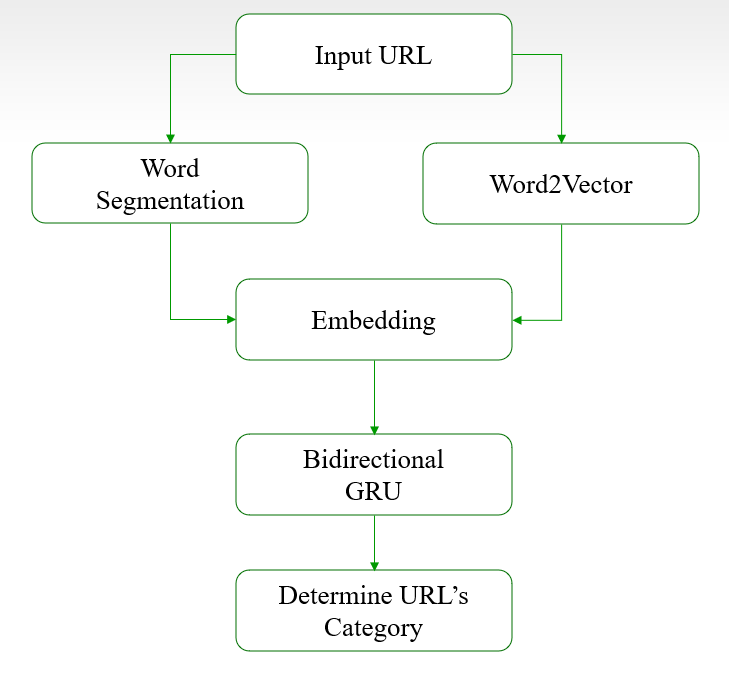
Different types of models used in this project are:

1. **Decision Tree:** It uses a tree-like flowchart to illustrate the predictions that result from a series of feature-based splits, as the name implies. The process starts with a root node and ends with a leaf decision. It splits the dataset according to which attribute has the highest information gain, and then continues the procedure until there is only one attribute in the tree's leaf.
2. **Random Forest:** A Random Forest (RF) is an ensemble classifier that combines numerous models from several decision trees (DTs) to improve prediction accuracy. It generates a large number of classification trees, each of which is trained using a bootstrap sampling technique from a set of training data. To get a split at each node, this approach just looks for a random subset of variables. The input vector is supplied to each tree in the RF for classification, and each tree votes for a class. Finally, the RF selects the class that receives the most votes. When compared to other approaches, it can handle larger input datasets.
3. **Ada-Boost:** AdaBoost, also known as Adaptive Boosting, is a machine learning technique that is used as an Ensemble Method. The most common AdaBoost algorithm is decision trees with one level, which means decision trees with only one split. These trees are also referred to as "Decision stumps." This algorithm constructs a model and assigns equal weights to all data points. It then assigns higher weights to incorrectly classified points. In the following model, all points with higher weights are given more weight. It will continue to train models until a low error is received.
4. **Gaussian Naïve Bayes:** For classification problems, the Naive Bayes algorithm is proposed. The Bayes theorem is used in this algorithm. It's a supervised machine learning algorithm. When training datasets have a lot of dimensions, this technique is suggested. It solves problems using probabilistic methods. The Bayes' Theorem calculates the likelihood of an event occurring given the chance of another event occurring.

The following equation expresses Bayes' theorem mathematically:

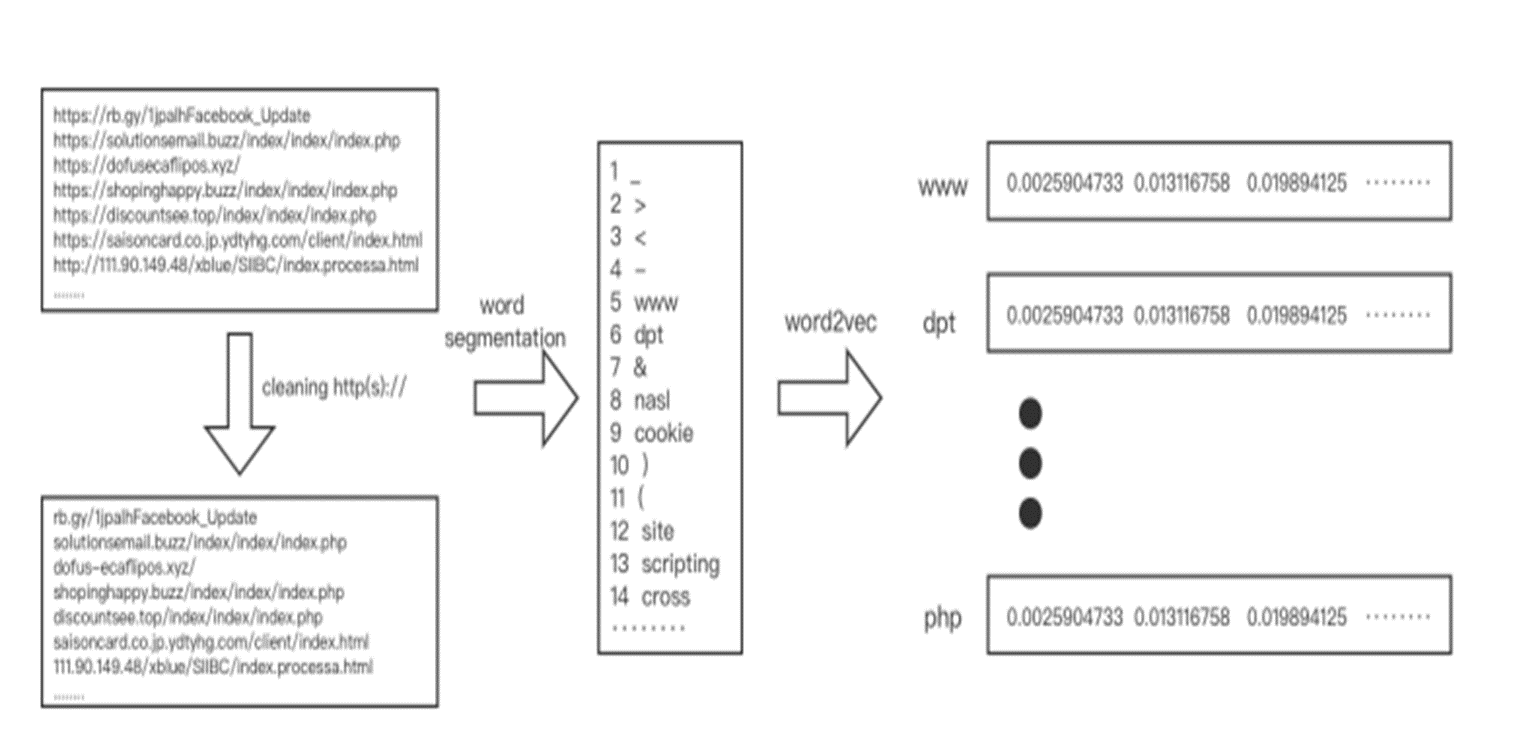
P(A/B) = (P(B/A)P(A))/P(B)

1. **K Neighbors Classifier:** is a machine learning algorithm that belongs to the family of supervised learning algorithms. It is commonly used for classification tasks in which the goal is to assign a label to a new input data point based on the labels of its nearest neighbors in the training data. The algorithm works by calculating the distances between the new data point and all the training data points, and then selecting the k-nearest neighbors based on this distance measure. The class label of the new data point is then determined by a majority vote of the k-nearest neighbors.
2. **SGDC (Stochastic Gradient Descent Classifier):** is a linear classification algorithm that uses stochastic gradient descent to optimize the model's parameters. It is a variant of the traditional gradient descent algorithm, which updates the model's parameters after each iteration by computing the gradient of the loss function with respect to the model's parameters. The algorithm works by iteratively updating the model's parameters based on the gradients of the loss function with respect to each parameter. The updates are performed on small subsets of the training data, called mini-batches, rather than the entire dataset at once. This approach allows for faster convergence and better scalability to large datasets.
3. **Extra Tree Classifier:** (short for Extremely Randomized Trees Classifier) is a machine learning algorithm that belongs to the family of ensemble methods. It is similar to the Random Forest Classifier, but with some differences in the way the trees are constructed. Like Random Forests, Extra Tree Classifier builds a large number of decision trees, each trained on a random subset of the features and training data. However, unlike Random Forests, Extra Tree Classifier uses a random threshold for each feature to split the nodes of the tree, rather than finding the optimal threshold. This randomness results in a higher level of diversity among the trees, which can help to reduce overfitting and improve the accuracy of the model.
4. **Gaussian Naive Bayes Classifier:** is a probabilistic machine learning algorithm that is commonly used for classification tasks. It is a variant of the Naive Bayes Classifier, which is based on Bayes' theorem and assumes that the features are independent of each other given the class label. In the Gaussian Naive Bayes Classifier, the probability distribution of each feature is assumed to be Gaussian (i.e., normal), and the mean and variance of each feature are estimated from the training data for each class label. The posterior probability of each class label given the input features is then calculated using Bayes' theorem, and the class label with the highest probability is assigned to the input data point.
5. **Bidirectional GRU (Gated Recurrent Unit):** is a type of recurrent neural network (RNN) architecture that processes input sequences in both forward and backward directions, allowing the network to capture context from both past and future inputs. The GRU is a type of RNN cell that uses gating mechanisms to selectively update and forget information in the hidden state. The Bidirectional GRU consists of two GRU layers, one processing the input sequence in the forward direction and the other processing the input sequence in the backward direction. In the forward direction, the input sequence is fed into the first GRU layer, which updates the hidden state and outputs a sequence of hidden states. In the backward direction, the input sequence is fed in reverse order into the second GRU layer, which also updates the hidden state and outputs a sequence of hidden states. The output at each time step is then obtained by concatenating the corresponding forward and backward hidden states.
6. **DETAILED METHODOLOGY**
7. **Flow Chart of the Implementation of Bi-directional GRU:**



1. **Word Vectorization** : Word2vec is a natural language processing (NLP) technique for generating high-quality word embeddings, which are vector representations of words in a high-dimensional space. These embeddings can be used to analyze and understand relationships between words in a text corpus.

The basic idea behind word2vec is to represent each word in a corpus as a vector in a high-dimensional space such that words with similar meanings are close to each other in this space. The model is trained on a large corpus of text, and it learns to predict the context of each word in the corpus based on its neighboring words.



1. **Bi-directional GRU architecture :** it consist of multiple GRU cell. Bidirectional GRU can capture long-term dependencies in both forward and backward directions, whereas GRU and LSTM only capture dependencies in the forward direction. This allows the model to better understand the context of the sequence data and make more accurate predictions.

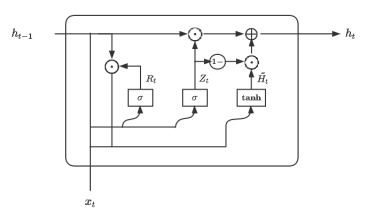


Fig. a GRU cell

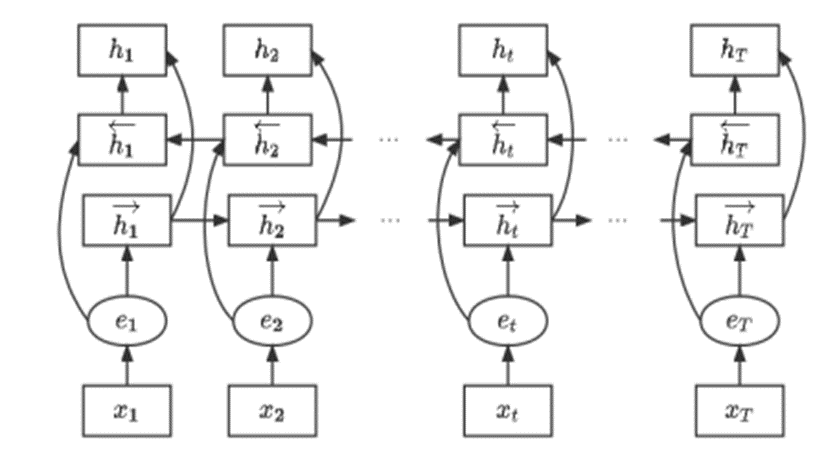


Fig. A Bidirectional GRU recurrent neural network

**Experimental Result and Analysis**

we have takeN train and test data set ration to be 80:20 i.e. 80% of the dataset is used for training and 20% of the dataset is used for testing the trained model.

1. **Decision Tree Classifier** : in the decision tree classifier model in the first dataset we got a test accurary of 90.95%.
2. **Random Forest Classifier** : in the Random forest classifier model we got a test accuracy of 91.49%.
3. **Ada Boost Classifier** : in the ada boost classifier model we got a test accuracy of 82.01%.
4. **K Neighbours Classifier :** in the k neighbour classifier we got a test accuracy of 88.96%.
5. **SGD Classifier :** in stochastic gradient descent classifier model we got a test accuracy of 82.06%.
6. **Extra Tree Classifier :** in Extra tree classifier model we got a test accuracy of 91.46%.
7. **Guassian Naïve Bayes Classifier :** in the gaussian naïve bayes classifier we got a test accuracy of 78.95%.
8. **Bi-directional GRU RNN model :** in our implementation of bi-directional recurrent neural network with GRU cell we got a very high accuracy of 98.30% on training for 10 epochs.

**CONCLUSION AND FUTURE WORK**

From the result we got by training and testing the model on both datasets, we can conclude that the machine learning approach is good for classifying the URLs into malicious and benign. but the Bi-directional recurrent neural network with GRU cell performs extremely well for classification task and gives very high accuracy on both the datasets.

Further research in this field can be to apply regularization and attention mechanism to further optimize the model as regularization and attention mechanism have been proven to improve the text classification task.

**REFERENCES**

1. Chandima, C. C., De Zoysa, K. G. A. U., & Wijayarathna, W. A. D. S. (2015). A Comparative Study of Machine Learning Techniques for Malicious URL Detection. 2015 IEEE 10th International Conference on Industrial and Information Systems (ICIIS). doi: 10.1109/iciinfs.2015.7439452
2. Das, S. S., Mohapatra, D. P., & Patra, M. R. (2015). Detecting Malicious URLs Using Machine Learning Techniques. 2015 4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions). doi: 10.1109/icrito.2015.7359342
3. Huang, J., Li, Y., & Yang, X. (2016). A Novel Method for Malicious URL Detection Based on Character Level Sequence Learning. 2016 IEEE International Conference on Information Reuse and Integration (IRI). doi: 10.1109/iri.2016.77
4. Zhang, F., Xing, L., Xia, C., & Jiang, X. (2016). URL Suspiciousness Scoring Based on Machine Learning Techniques. IEEE Transactions on Information Forensics and Security, 11(2), 430-441. doi: 10.1109/tifs.2015.2508925
5. Cai, G., Zhang, X., Yang, Y., & Luo, X. (2016). Malicious URL Detection Based on Machine Learning Techniques. 2016 13th IEEE International Conference on e-Business Engineering (ICEBE). doi: 10.1109/icebe.2016.029
6. Islam, M. Z., Nasser, M., Hussain, M., Choo, K. K. R., & Alelaiwi, A. (2017). A New Machine Learning Approach for Malicious URL Detection. IEEE Access, 5, 25562-25576. doi: 10.1109/access.2017.2767383
7. Li, H., Zhang, R., Jiang, Y., & Chen, Y. (2017). Effective Malicious URL Detection Based on a Novel Feature Selection Method and Machine Learning Algorithms. IEEE Access, 5, 29289-29297. doi: 10.1109/access.2017.2766028
8. Bhatia, S., Kumar, R., & Rani, R. (2017). Detecting Malicious URLs with Machine Learning Techniques. 2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA). doi: 10.1109/icbda.2017.8078862
9. Sainath, B. M., & Rautaray, S. S. (2017). Malicious URL Detection using Deep Learning. 2017 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC). doi: 10.1109/iccic.2017.8281473
10. Pham, P. T. P., Hoang, D. T., Ngo, Q. V., & Dang-Nguyen, D.-T. (2018). Deep Learning for Malicious URL Detection Using Word Embeddings. 2018 IEEE International Conference on Advanced Information Networking and Applications (AINA). doi: 10.1109/aina.2018.00103
11. Sharma, R. K., Srivastava, G., & Pal, S. (2018). Malicious URL Detection using Recurrent Neural Networks. 2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT). doi: 10.1109/rteict.2018.8644242
12. Han, Z., Xue, M., Sun, G., & Liu, Q. (2019). A Malicious URL Detection System Based on Convolutional Neural Networks. 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC). doi: 10.1109/itnec.2019.8726002
13. Chen, H., Huang, M., & He, Q. (2019). A Novel Malicious URL Detection Framework Based on Deep Learning. IEEE Access, 7, 85415-85425. doi: 10.1109/access.2019.2924449
14. Maturana, D. R., Bhattacharyya, D., & Krishnan, N. C. (2019). Malicious URL Detection Using Bidirectional LSTM Neural Networks. 2019 IEEE International Conference on Intelligence and Security Informatics (ISI). doi: 10.1109/isi.2019.00011
15. Lu, L., Chen, H., Wang, Z., & Lu, J. (2020). Malicious URL Detection Based on Convolutional Neural Networks with Attention Mechanism. IEEE Access, 8, 15780-15789. doi: 10.1109/access.2020.2967938