



KNOWLEDGE-BASED APPROACH TO DETECT POTENTIALLY RISKY WEBSITES

Mrs.R.L.Indu Lekha, M.E¹, Chaithra N², Deepa Sri³, Kamna Agarwal⁴

Assistant Professor, Department of Computer Science and Engineering, Adhiyamaan College of Engineering,
Tamil Nadu, India¹

UG Scholar, Department of Computer Science and Engineering, Adhiyamaan College of Engineering,
Tamil Nadu, India²⁻⁴

Abstract-Malicious website causes huge money losses and irreparable damage for companies and particulars. To face this situation, governments have approved multiple law projects. The first component is a previously built knowledge base and the second one complements the system with a binary classifier. In this project, we describe an approach to this problem based on automated URL classifications, using statistical methods. The proposed system uses the logistic regression and host-based properties of malicious website URLs. These methods are highly predictive models be extracting and automatically analyzing features of suspicious URLs. This program will predict the malicious website as a CSV file in the database and that risky website information will be sent to the cyber security department by using SMTP protocol.

INTRODUCTION

The Web has become a platform for supporting a wide range of criminal enterprises such as spam-advertised commerce (e.g., counterfeit watches or pharmaceuticals), financial fraud, and as a vector for propagating malware (e.g., so-called “drive-by downloads”). The common thread among the people is the requirement that unsuspecting users visit their sites. These visits can be driven by email, web search results, or links from other web pages, but all require the user to take some action. If one could inform users beforehand that a particular URL was dangerous to visit, much of this problem could be alleviated. To this end, the security community has responded by developing blacklisting services encapsulated in toolbars, appliances, and search engines that provide precisely this feedback. Inevitably, many malicious sites are not blacklisted either because they are too new, were never evaluated, or were evaluated incorrectly. To address this problem, with the help of machine learning he tried to find out malicious websites which are very harmful to browse in our system. As we use a certain algorithm to train them to find the harmful viruses which attack the user and prevent them by alerting the user not to access the site which he is trying to browse.

OBJECTIVE

- Public bodies that prosecute fraudulent and malicious websites dedicate a significant amount of time and resources to detect spam and malware on the internet.
- Most of this work is usually manual, which translates into hard and inefficient efforts.
- For this reason, it has become essential to develop systems able to automate the classification of websites into potentially risky or non-risky according to the features of these sites.
- The risky website information will be informed to the cyber security department by using SMTP protocol.

II.LITERATURE SURVEY

[1] A. Ali Ahmed, “Malicious Website detection: A review,”2018. Most of this work is usually manual, which translates into hard and inefficient efforts. For this reason, it has become essential to develop systems able to automate the classification of websites into potentially risky or non-risky according to the features of these sites. In this context, a risky website is one with malicious, unsafe, or fraudulent content with dangerous intentions against its visitors.

[2] A. Fernandez-Isabel, J. C. Prieto, F. Ortega, I. Martin de Diego, J. M. Moguerza, j. Mena, S. Galindo, and L. Napalkova,” A unified knowledge compiler to provide support the scientific community,”2018. KBSs are frameworks able to process data and information in order to generate knowledge using Artificial Intelligence (AI) to solve general tasks. These systems usually comprehend a storage component (e.g., a database) to ease the knowledge retrieval in response to specific queries, along with learning and justification, or to transfer knowledge from one domain of knowledge to another. They are formed by different modules to address the needs of the users or to optimize the system.

[3] Y. Zhauniarovich, I. Khalil, T. Yu, and M. Dacier,” A survey on malicious domains detection through DNS data analysis,”2019. There are similar studies that use the domain to detect malicious websites. However, these studies usually use textual features or they use an IP address approach. Other alternatives use the Domain Name System(DNS).



[4] M. Ferreira, "Malicious URL Detection using Machine Learning Algorithms," pp.114-122,2019. In this process, we train the machine learning algorithms using python to find the malicious websites which can be used in the browser we use. When we type the web address in the address bar of the browser we use algorithms that we trained that will automatically do their work by checking the web address if it's a safe site it will process by browsing the website and if it is a malicious URL then it will display a dialog box in which it says a message that it is an unprotected malicious website do u still want to process.

[5] M. A. Ferran, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," J. Inf. Secure. Appl., vol. 50, p. 102419, 2020. Effectiveness of the phishing URL detection approach. Ablation experiment Character-level spatial feature representation extracted word-level temporal feature representation from character-level CNNs and attention-based URLs Hierarchical RNN modules improve performance, the generalization of this approach.

III.EXISTING METHOD & PROPOSED METHOD

EXISTING METHOD:

- Existing system SVM-based model systematically categorizes the various types of feature representation used for creating the training data for this task and also categorizes various learning algorithms used to learn a prediction model rate low compared to proposed model logistic regression.
- Blacklists are essentially a database of URLs that have been confirmed to be malicious in the past. This database is compiled over time often through crowd-sourcing solutions complex to analyze the existing system.
- Blacklisting methods, thus have severe limitations, and it appears almost trivial to bypass them, especially because blacklists are useless for making predictions on new URLs.

PROPOSED SYSTEM:

In this project, we use a machine learning-based to classify the risky, non-risky, and neutral domain websites by using a logistic regression algorithm. Where algorithm goal is to find out whether the current URL website which is browsing is good to go for browsing or bad. By using the machine learning algorithm, we are going to classify which type of URL is a suspicious URL. For that detection process, we have a data set about URL from GUI which we use for the processing and extracting the feature and through the data set we used for the processing and a huge dataset which is classified over, the data set is split for training and testing data are in ratio 75/25. The data set is trained and after the training and accuracy obtain later that passing the test dataset and predicting the result for the test dataset final we use the new data for the prediction of unknown data and the result is obtained. If the machine learning algorithm predicts the malicious website the CSV file can be stored in the MySQL database and that risky website information will be informed to the cyber security department by using SMTP protocol.

IV.SYSTEM FUNCTION

MODULES

- Input Interface Module
- Domains Extraction Module
- Prediction Module
- Classification Module
- Information Updating Module

1. INPUT INTERFACE MODULE:

In this module, the input URL website domains can be interfaced with software for analysis. The input URL website can be interfaced by using the pyqt5 front end page.

2. DOMAINS EXTRACTION MODULE:

The unstructured information about the URL (e.g., textual description) is appropriately formatted and converted to a numerical vector so that it can be fed into machine learning algorithms. For malicious URL detection, we have proposed several types of features that can be used to provide useful information. We categorize these features into Blacklists Features, URL-based Lexical Features, Host-based features, Content-based Features, and others (Context and Popularity).

3. PREDICTION MODULE:

In this module, the machine learning algorithm will predict the non-risky website based on the logistic regression algorithm. This algorithm will be used the training dataset as a reference.

4. CLASSIFICATION MODULE:

In this module, the risky, neutral, and non-risky websites can be classified by using comparing with the database training module. The non-risky website data can be stored in MySQL (CSV file) database.

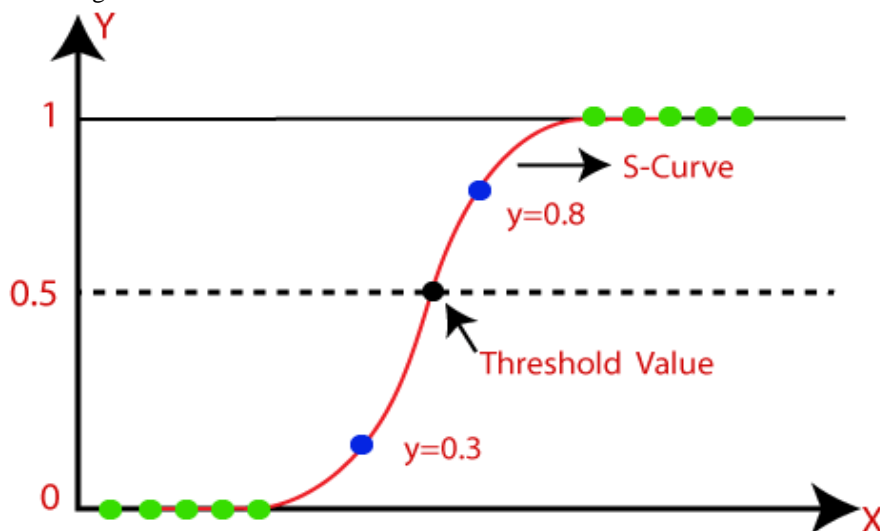


5. INFORMATION UPDATING MODULE:

If the machine learning algorithm predicts the malicious website the CSV file can be stored in the MySQL database and that risky website information will be informed to the cyber security department by using SMTP protocol.

LOGISTIC REGRESSION ALGORITHM

- Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.
- Logistic regression predicts the output of a categorical dependent variable. Therefore, the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or false, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.
- Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems.
- In Logistic regression, instead of fitting a regression line, we fit an “S” shaped logistic function, which predicts two maximum values(0 or 1).
- The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc.
- Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets.



V.SYSTEM SPECIFICATION

HARDWARE SPECIFICATION

Processor type : i5 processor
 RAM : 8GB RAM, 64 bit
 Storage : 1TB
 Display : 20' color display

SOFTWARE SPECIFICATION

Front end : pyqt5
 Back end : Python 3.9
 Software tool used : PyCharm
 Platform : Windows 8

VI.SYSTEM SOFTWARE

PYCHARM

PyCharm is a dedicated Python Integrated Development Environment (IDE) providing a wide range of essential tools for Python developers, tightly integrated to create a convenient environment for productive Python, web, and data science development.



PyCharm of Features

1. Supported Platforms

PyCharm is a cross-platform IDE that works on Windows, macOS, and Linux. Check the system requirements: We can install PyCharm using Toolbox or standalone installations. Start with a project in PyCharm. Everything we do in PyCharm, we do within the context of a project. It serves as a basis for coding assistance, bulk refactoring, coding style consistency, and so on. We have three options to start working on a project inside the IDE:

- Open an existing project
- Check out a project from version control
- Create a new project

Open an existing project

Begin by opening one of our existing projects stored on our computer. We can select one in the list of the recent projects on the Welcome screen or click open.

2. PyCharm Window Page

Otherwise, we can create a project for our existing source files. Select the command Open on the File menu and specify the directory where the sources exist. PyCharm will then create a project from sources for us and then enter our credentials to access the storage. Then, enter a path to the sources and clone the repository to the localhost.

3. PyCharm Window Creating New Project

When creating a new project, we need to specify a Python interpreter to execute Python code in our project. We need at least one Python installation to be available on our machine. For a new project, PyCharm creates an isolated virtual environment: venv, pipenv, or Conda. As we work, we can change it or create new interpreters. We can also quickly preview packages installed for our interpreters and add new packages in the Python Package tool window. When we launch PyCharm for the very first time, or when there are no open projects, we see the Welcome screen. It gives us the main entry points into the IDE: creating or opening a project, checking out a project from version control, viewing documentation, and configuring the IDE.

VII.HARDWARE REQUIREMENTS & SPECIFICATIONS

I5 PROCESSOR

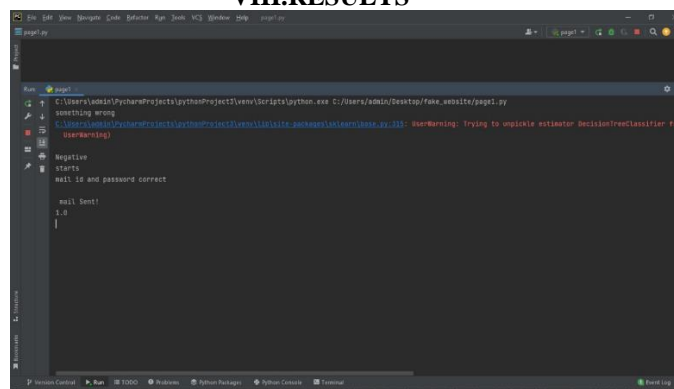
The core is a family of I5 processors famous for its innovative design and integrated architecture that also offer the same computer benefits. is also good at providing users with excellent user interactive images.

Basic Features of the I5

The basic features of the I5 features are significantly improved compared to the previous Intel version. Some of the most popular and advanced features of I4 processors are listed below.

- Intel I5 processor is fully loaded with the latest HD graphics with a powerful and advanced video engine that provides a smooth high-quality display and 3d graphical capability. All I5 processors can be considered high-end image processors and everyday computer multimedia.
- Intel I5 processors also provide hyper-threading technology to its users that allow for multiple capabilities for both the user and the system. Systems with I5 processors can perform and integrate two tasks simultaneously without causing performance delays and bug fixes. They are so responsive that the exit of the systems can be done at the same time as well. we can easily say that the Intel I5 is the best choice for homes and offices. More than seven applications can run simultaneously on the system with an I5 processor built into the motherboard

VIII.RESULTS





IX.CONCLUSION

Malicious URL detection plays a critical role in many cyber security applications, and clearly, machine learning approaches are a promising direction. In this project, we proposed the logistic regression algorithm for Malicious URL Detection using machine learning techniques. In particular, we offered a systematic formulation of Malicious URL detection from a machine learning perspective, and then detailed the discussions of existing studies for malicious URL detection, particularly in the forms of developing new feature representations and designing new learning algorithms for resolving the malicious URL detection tasks. In this project, we categorized most, if not all, the existing contributions for malicious URL detection in literature, and also identified the requirements and challenges for developing Malicious URL Detection as a service for real-world cyber security applications

X.REFERENCES

- [1]. A. I. Schein and L. H. Ungar, Active learning for logistic regression: An evaluation, vol. 68, no. 3. 2017.
- [2]. M. R. Segal, "Machine learning benchmarks and random forest regression. Center for Bioinformatics and Molecular Biostatistics, UC San Francisco, USA.,"2018.
- [3]. F. Livingston, "Implementation of Breiman's Random Forest Machine Learning Algorithm," Mach Learn. J. Pap., pp. 1-13, 2019.
- [4]. M. W. Gardner and S. R. Dorling, "Artificial neural (the multilayer perceptron) – a review of applications in the atmospheric sciences," *Atmos. Environ.*, vol 32, no. 14-15, pp. 2627-2636, 2020.
- [5]. J. Burrell, "How the machine 'think': Understanding opacity in machine learning algorithms," *Big Data Soc.*, vol. 3, no. 1, pp. 1-12, 2019.
- [6]. S. Dreiseitl and L. Ohno-Machado, "Logistic regression and artificial neural network classification models: A methodology review," *J. Biomed. Inform.*, vol. 35, no. 5-6, pp. 352-359, 2020.
- [7]. X. Yan, Y. Xu, B. Cui, S. Zhang, T. Guo, and C. Li, "Learning URL Embedding for Malicious Website Detection," *IEEE Trans. Ind. Informatics*, vol. 3203, no. c, pp. 1-1, 2020.
- [8]. M. Alazab and S. Fellow, "Malicious URL Detection using Deep Learning," pp. 1-9, 2020.
- [9]. Y. L. Zhang et al., "Poster: A PU learning-based system for potential malicious URL detection," *Proc. ACM Conf. Comput. Commun. Secure.*, pp. 2599-2601, 2017.
- [10]. Y. Huang, Q. Yang, J. Qin, and W. Wen, "Phishing URL detection via CNN and attention-based hierarchical RNN," *Proc. -2019 18th IEEE Int. Conf. Trust. Secure. Priv. Comput. Commun. IEEE Int. Conf. Big Data Sci. Eng. Trust. 2019*, pp. 112-119, 2019.
- [11]. M. A. Ferran, L. Maglaras, S. Moschogiannis, and H. Janicke, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," *J. Inf. Secure. Appl.*, vol. 50, p. 102419, 2020.
- [12]. R. Vijayakumar, K. P. Soman, P. Poorna Chandran, V. S. Mohan, and A. D. Kumar, "Scale Net: Scalable and hybrid framework for cyber thread situational awareness based on DNS, URL, and email data analysis," *J. Cyber Secure. Mobil.*, vol. 8, no. 2, pp. 189-240, 2018.
- [13]. O. K. Sahingoz, E. Buber, O. Demir, and B. Diri, "Machine learning-based phishing detection from URLs," *Expert Syst. Appl.*, vol. 117, pp. 345-357, 2019.
- [14]. J. Ma, L. K. Saul, S. Svage, and G. M. Voelker. Identifying Suspicious URLs: An Application of Large-Scale Online Learning. In *Proc. of the International Conference on Machine Learning(ICML)*, Montreal, Quebec, June 2018.
- [15]. B. Scholkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, 2020.