Detection of Malicious URLs using Machine Learning Techniques

C M Kesava Kumar

Department of Computer Science, Sri Venkateswara University, Tirupati

***Abstract*—** *Internet surfing has become a vital part of our daily life. So, to catch the attention of the users' different browser vendors compete to set up the new functionality and advanced features that become the source of attacks for the intruder and the websites are put at hazard. However, the existing approaches are not adequate to protect the surfers which require an expeditious and precise model that can be able to distinguish between the benign or malicious webpages. In this research article, we design a new classification system to analyse and detect the malicious web pages using machine learning classifiers such as, random forest, support vector machine. naïve Bayes, logistic regression and Some special URL (Uniform Resource Locator) based on extricated features the classifiers are trained to predict the malicious web pages. The experimental results have shown that the performance of the random forest classifier achieves better accuracy of 95% in comparison to other machine learning classifiers.*

I. INTRODUCTION

With the rapid development of the web, more and more services like internet banking, e-commerce, social networking, shopping, making a bill payment, e-learning, etc. are available to users and they are surfing the internet via browsers or web application. As the browsers are come up with different advanced features and functionalities which leads to risk by losing their personal and sensitive information. As the naïve users are not aware of the different malware so they are easily trapped by the intruder by just a single click on the malicious web sites which allows the invaders to detect the vulnerabilities on the web page and inject the payloads to get remote access to victim's web page.

Therefore, the precise identification of web pages in an ever-growing web environment is very important. Blacklisting services were embedded in the browsers to face the challenges but it has several disadvantages like incorrect listing. In this article, we explore a self-learning approach to classify the web page based on a small feature set. We use four machine learning classifiers to classify the web site into two classes benign and malicious web pages. "The rest of the research work is planned as follows: Section II presents related work, the methodology is discussed in section III, experimental result analysis is depicted in Section IV and Section V contains the conclusion of the research work and suggests some future work".

1.1 Artificial Intelligence:

Artificial intelligence (AI) is the ability of a computer program or a machine to think and learn. It is also a field of study which tries to make computers "smart". As machines become increasingly capable, mental facilities once thought to require intelligence are removed from the definition. AI is an area of computer sciences that emphasizes the creation of intelligent machines that work and reacts like humans. Some of the activities computers with artificial intelligence are designed for include: Face recognition, Learning, Planning, Decision making etc.,

Artificial intelligence is the use of computer science programming to imitate human thought and action by analysing data and surroundings, solving or anticipating problems and learning or self-teaching to adapt to a variety of tasks.

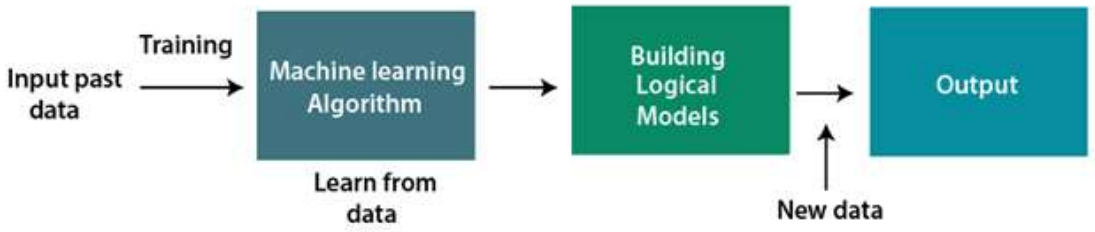
1.2 Machine Learning

Machine learning is a growing technology which enables computers to learn automatically from past data. Machine learning uses various algorithms for **building mathematical models and making predictions using historical data or information.** Currently, it is being used for various tasks such as **image recognition, speech recognition, email filtering, Facebook auto-tagging, recommender system**, and many more.

Machine Learning is said as a subset of **artificial intelligence** that is mainly concerned with the development of algorithms which allow a computer to learn from the data and past experiences on their own. The term machine learning was first introduced by **Arthur Samuel** in **1959**. We can define it in a summarized way as: "Machine learning enables a machine to automatically learn from data, improve performance from experiences, and predict things without being explicitly programmed".

A Machine Learning system **learns from historical data, builds the prediction models, and whenever it receives new data, predicts the output for it**. The accuracy of predicted output depends upon the amount of data, as the huge amount of data helps to build a better model which predicts the output more accurately.

Suppose we have a complex problem, where we need to perform some predictions, so instead of writing a code for it, we just need to feed the data to generic algorithms, and with the help of these algorithms, machine builds the logic as per the data and predict the output. Machine learning has changed our way of thinking about the problem. The below block diagram explains the working of Machine Learning algorithm:



1.2.1 Features of Machine Learning:

- Machine learning uses data to detect various patterns in a given dataset.
- It can learn from past data and improve automatically.
- It is a data-driven technology.
- Machine learning is much similar to data mining as it also deals with the huge amount of the data.

1.2.2 Classification of Machine Learning

At a broad level, a model using labelled data to understand the datasets and learn about each data, once the training and processing are done then we test the model by providing a sample data to check whether it is predicting the exact output or not.

The goal of supervised learning is to map input data with the output data. The supervised learning is based on supervision, and it is the same as when a student learns things in the supervision of the teacher. The example of supervised learning is **spam filtering**.

Supervised learning can be grouped further in two categories of algorithms:

- **Classification**
- **Regression**

2) Unsupervised Learning

Unsupervised learning is a learning method in which a machine learns without any supervision. The training is provided to the machine with the set of data that has not been labelled, classified, or categorized, and the algorithm needs to act on that data without any supervision. The goal of unsupervised learning is to restructure the input data into new features or a group of objects with similar patterns.

In unsupervised learning, we don't have a predetermined result. The machine tries to find useful insights from the huge amount of data.

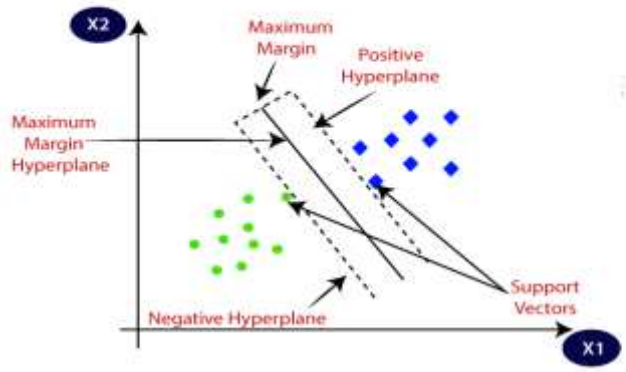
It can be further classifieds into two categories of algorithms:

- **Clustering**
- **Association**

1.3 Support Vector Machine (Svm)

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning. The goal of

the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.



SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:

Hyperplane: There can be multiple lines/decision boundaries to segregate the classes in n-dimensional space, but we need to find out the best decision boundary that helps to classify the data points. This best boundary is known as the hyperplane of SVM.

The dimensions of the hyperplane depend on the features present in the dataset, which means if there are 2 features (as shown in image), then hyperplane will be a straight line. And if there are 3 features, then hyperplane will be a 2-dimension plane.

We always create a hyperplane that has a maximum margin, which means the maximum distance between the data points.

Support Vectors: The data points or vectors that are the closest to the hyperplane and which affect the position of the hyperplane are termed as Support Vector. Since these vectors support the hyperplane, hence called a Support vector.

1.4 Naive Bayes

1. Naïve Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems.
2. It is mainly used in *text classification* that includes a high-dimensional training dataset.
3. Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.
4. It is a probabilistic classifier, which means it predicts on the basis of the probability of an object.
5. Some popular examples of Naïve Bayes Algorithm are spam filtration, Sentimental analysis, and classifying articles.

The Naïve Bayes algorithm is comprised of two words Naïve and Bayes, Which can be described as:

- **Naïve**: It is called Naïve because it assumes that the occurrence of a certain feature is independent of the occurrence of other features. Such as if the fruit is identified on the bases of colour, shape, and taste, then red, spherical, and sweet fruit is recognized as an apple. Hence each feature individually contributes to identify that it is an apple without depending on each other.
- **Bayes**: It is called Bayes because it depends on the principle of Bayes' Theorem.

Bayes' theorem

- Bayes' theorem is also known as **Bayes' Rule** or **Bayes' law**, which is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability.

- The formula for Bayes' theorem is given as:

$$P(A|B) = \frac{P(B \setminus A)P(A)}{P(B)}$$

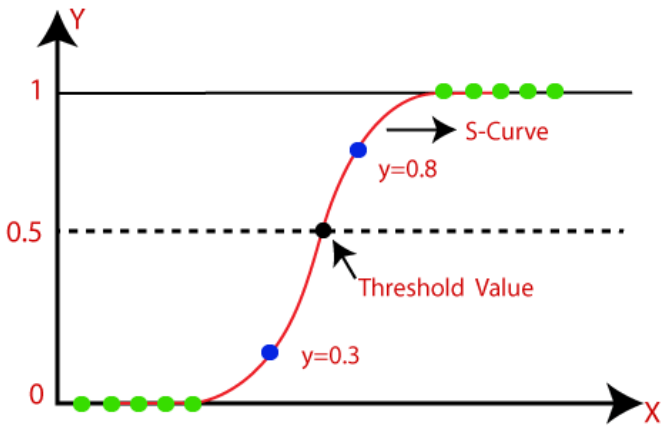
Where, P(A|B) is Posterior probability: Probability of hypothesis A on the observed event B.

1.5 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. Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification. The below image is showing the logistic function.



1.5.1 Logistic Function (Sigmoid Function):

- The sigmoid function is a mathematical function used to map the predicted values to probabilities.
- It maps any real value into another value within a range of 0 and 1.
- The value of the logistic regression must be between 0 and 1, which cannot go beyond this limit, so it forms a curve like the "S" form. The S-form curve is called the Sigmoid function or the logistic function.
- In logistic regression, we use the concept of the threshold value, which defines the probability of either 0 or 1. Such as values above the threshold value tends to 1, and a value below the threshold values tends to 0.

1.6 Random Forest Algorithm

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of **ensemble learning,** which is a process of *combining multiple classifiers to solve a complex problem and to improve the performance of the model.*

As the name suggests, **"Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset."** Instead of relying on one decision

tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

Below are some points that explain why we should use the Random Forest algorithm:

- It takes less training time as compared to other algorithms.
- It predicts output with high accuracy, even for the large dataset it runs efficiently.
- It can also maintain accuracy when a large proportion of data is missing.

Random Forest works in two-phase first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase.

The Working process can be explained in the below steps and diagram:

- **Step-1:** Select random K data points from the training set.
- **Step-2:** Build the decision trees associated with the selected data points (Subsets).
- **Step-3:** Choose the number N for decision trees that you want to build.
- **Step-4:** Repeat Step 1 & 2.
- **Step-5:** For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

II. LITERATURE SURVEY

[1] TITLE: Context-sensitive and keyword density-based supervised machine learning techniques for malicious webpage detection.

AUTHOR: Altay, Betel, Tansel Dokeroglu, and Ahmet Cosar – 2019

DESCRIPTION: Conventional malicious webpage detection methods use blacklists in order to decide whether a webpage is malicious or not. The blacklists are generally maintained by third-party organizations. However, keeping a list of all malicious Web sites and updating this list regularly is not an easy task for the frequently changing and rapidly growing number of webpages on the web. In this study, we propose a novel context-sensitive and keyword density-based method for the classification of webpages by using three supervised machine learning techniques, support vector machine, maximum entropy, and extreme learning machine. Features (words) of webpages are obtained from HTML contents and information is extracted by using feature extraction methods: existence of words, keyword frequencies, and keyword density techniques. The performance of proposed machine learning models is evaluated by using a benchmark data set which consists of one hundred thousand webpages. Experimental results show that the proposed method can detect malicious webpages with an accuracy of 98.24%, which is a significant improvement compared to state-of-the-art approaches.

[2] TITLE: "WebMon: ML-and YARA-based malicious webpage detection."

AUTHOR: Kim, Sungjin, Jinkook Kim, Seokwoo Nam, and Dohoon Kim. - 2018

DESCRIPTION: Attackers use the openness of the Internet to facilitate the dissemination of malware. Their attempts to infect target systems via the Web have increased with time and are unlikely to abate. In response to this threat, we present an automated, low-interaction malicious webpage detector, WebMon, that identifies invasive roots in Web resources loaded from WebKit2-based browsers using machine learning and YARA signatures. WebMon effectively detects hidden exploit codes by tracing linked URLs to confirm whether the relevant websites are malicious. WebMon detects a variety of attacks by running 250 containers simultaneously. In this configuration, the proposed model yields a detection rate of 98%, and is 7.6 times faster (with a container) than previously proposed models. Most importantly, Web Môn's focus on extracting malicious paths in a domain is a novel approach that has not been explored in previous studies.

[3] TITLE: "Detection of malicious web pages based on hybrid analysis."

AUTHOR: Wang, Rong, Yan Zhu, Jiefan Tan, and Binbin Zhou. - 2017

DESCRIPTION: Malicious web pages have become an increasingly serious threat to web security in recent years. In this paper, we propose a new detection method that consists of static and dynamic analyses for detecting malicious web pages. Static analysis utilizes classification algorithms in machine learning to identify certain benign and malicious web pages. As a complement to static analysis, dynamic analysis mainly checks the unknown web pages to determine whether they have malicious shellcodes during their execution. Because of the combination of static and dynamic analyses, the proposed detection method achieves high performance, and it has a light weight and is simple to use.

[4] TITLE: "Machine Learning Classifiers to Detect Malicious Websites."

AUTHOR: Urcuqui, Christian, Andres Navarro, Jose Osorio, and Melisa Garcia. - 2017

DESCRIPTION: A risk that exists in Internet is the access of websites with malicious content, because they might be open doors for cybercrimes or be the mechanism to download files in order to affect organizations, persons and the environment. What is more, the attack registers through websites have been part of cyberattacks reports during the last years; this information includes attacks made by the currently risks found in new technologies, such as the IoT. Due the computer security complexity, studies have been working in to use machine learning algorithms to identify web malicious content. This article explores the application of a data analysis process through a framework that includes dynamic, static analysis, updated websites and a low interaction client honeypot in order to classify a website. Furthermore, it evaluates the capacity of the classification of four machine learning through the information analysed.

[5] TITLE: "Two-phase malicious web page detection scheme using misuse and anomaly detection."

AUTHOR: Yoo, Suyeon, Sehun Kim, Anil Choudhary, O. P. Roy, and T. Tuithung. - 2014

DESCRIPTION: Misuse detection method and anomaly detection method are widely used for the detection of malicious web pages. Both are based on machine learning. Misuse detection can detect known malicious web pages, but it cannot detect new ones. In contrast, anomaly detection can detect unknown malicious web pages, but it has a high false positive rate. In order to achieve a high detection rate through precisely detecting known and unknown malicious web pages, we propose a two-phase detection scheme. In the first phase, the misuse detection model is built based on the C4.5 decision tree algorithm, which allows known malicious web pages to be detected. In the second phase, the anomaly detection model with a one-class support vector machine is used to detect new types of malicious web pages. The experimental results show that our proposed method has significantly higher malicious web page detection rate than conventional ones with the expense of slightly higher false positive rate.

III. PROBLEM STATEMENT

Predicting if a given website is malicious or benign using machine learning classifiers are logistic regression, random forest, naive Bayes and SVM.

The Existing system proposed a method for classifying malicious web pages using 30 features with the help of machine learning algorithm K-NN and SVM. The result of K-NN was better than SVM. Two classification models were used for detecting the malicious web pages and specific threat types.

Two types of detection methods: misuse detection and anomaly detection for identifying known and unknown malicious web pages respectively. Though the detection rate was relatively high up to 98.9% it's the false positive rate was high which is 30.5%. They have conducted their experiment in WEKA tool with dataset RafaBot.

Developed an interaction tool, Spider Net which was able to detect the malicious web page. The tool was implemented in MatLab. Two machine learning classifiers, multi-SVM, and ELM were implemented in the tool by taking three feature sets namely common features, redirect features, JavaScript features, and XSS attack features showing higher accuracy in ELM (96.62%) than multi-SVM (93.22%).

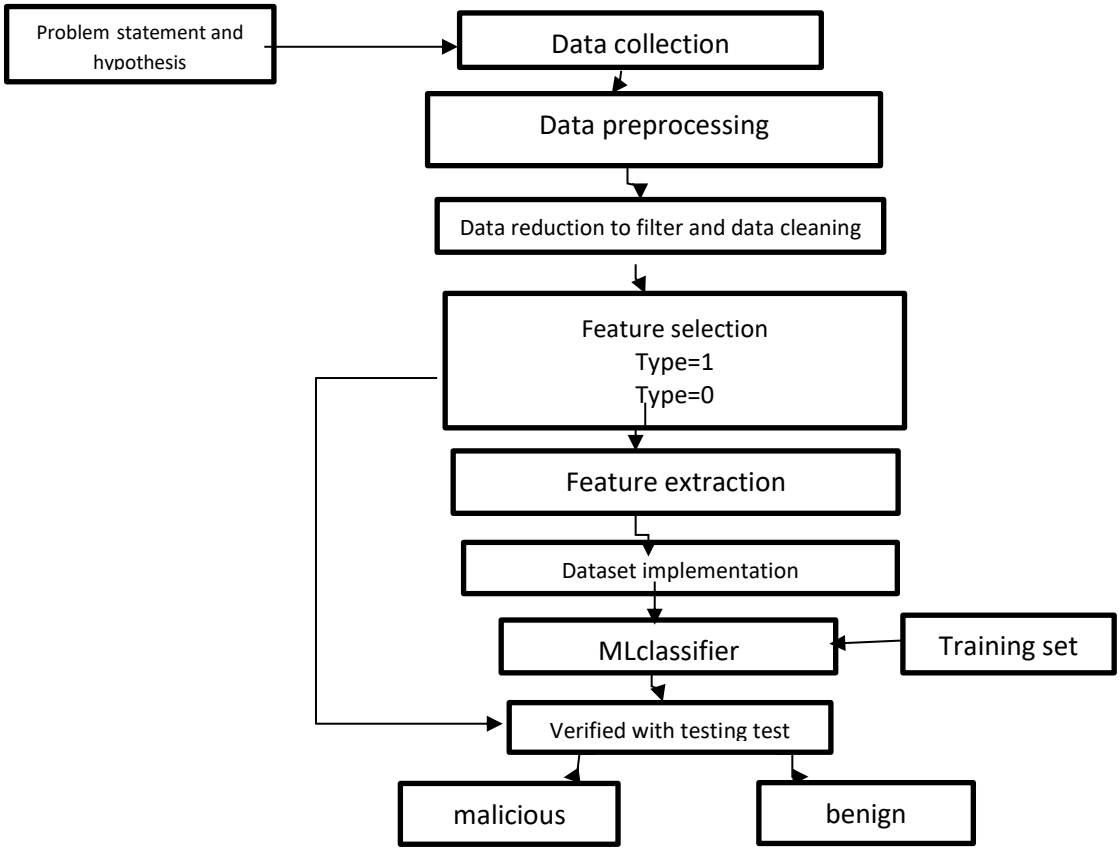
IV. PROPOSED SYSTEM

Our method uses a variety of discriminative features including textual properties, link structures, webpage contents, DNS information, and network traffic. Many of these features are novel and highly effective. Our experimental studies with 40,000 benign URLs and 32,000 malicious URLs obtained from real-life Internet sources show that our method delivers a superior performance: the accuracy was over 98% in detecting malicious URL sand over 93%inidentifying attack types. Wealsoreport our studies on the effectiveness of each group of discriminative features, and discuss their evadability

Advantages:

- ✓ accuracy is maximum
- ✓ Prediction is accurate

V. SYSTEM ARCHITECTURE:



Proposed implementation

In this section, we provide a detailed discussion about our proposed approach to identifying the malicious web page. To address the drawback of previous studies we design a new web site classification system based on the URL features to identify malicious websites.

In step 1 according to our requirements, we have collected a dataset from the internet source contains both the malicious and benign web sites. In step 2 data is reduced to filter and data cleaning by selecting a few relevant attributes out of 21 attributes in total from the dataset. In step 3 we have designed our dataset consisting of 7 URL features and 1782 records. Then we manually divide the dataset into two sets; one is for training set made up of 812 records and another is for testing set consists of 970 records. In step 4 machine learning classifiers are trained to create a Machine Learning (ML) model with the help of the training set. In the final step, the ML model is verified with the testing set to obtain our required result. If the Type attribute value contains 0 means the inputted URL is a benign web site else it is a malicious web site. The following subsection explains the three basic components of our approach: the dataset, feature extraction, and machine learning classifiers elaborately.

Dataset

The selection of datasets has a great influence on the quality of classification. We require to select an appropriate dataset as well as possible So we fill this gap by collecting a URL dataset from the Kaggle database [14] which is composed of both malicious and benign websites and it has 1782 records and 21 features. Out of 1782 records, 812 records are used. A snapshot of our dataset is depicted

Feature Extraction

Various methodologies are used in extracting the features. In our research work we have extracted the features manually based on the URL because in some cases by looking into the URL we can identify the maliciousness of web pages to some extent or by querying the information associated with the referenced host, it's safety can be detected. The advantage of using URL characteristics is that it avoids downloading the actual web page content also it can adapt to more environments including web pages and emails. We extract 7 essential syntactical and hosted features of URL, out of 21 features from the dataset. The two most essential features like SOURCE-APP-PACKETS and REMOTE-APP-PACKETS are added in our feature list which creates a huge difference in the detection of maliciousness of web pages in comparison to other attributes[14], which is represented in Fig.3 and these two attributes are not used in any of the existing approaches. Hence our classification approach is proved to be better than the existing approaches. The selected URL features are listed in Table II. Our proposed detection approach uses the machine learning algorithms and it requires these URL based features to distinguish the malicious web page from the benign web pages. the relationship between attributes and the type of web site is represented. APP_PACKETS consists of two attributes SOURCE-APP-PACKETS and REMOTEAPP-PACKETS. These two attributes show a huge difference in identifying malicious and benign web sites. The degree of maliciousness is more for this attribute which is plotted in Fig.3 hence more risk to web sites. Therefore, we have added these two features in our feature list.

Machine Learning Classifiers

There are a lot of methods for realizing the classifiers. We select four machine learning algorithms to build our classifiers "Logistic Regression is a supervised machine learning technique. Random Forest is an ensemble learning method Gaussian Naïve Bayes is a simple, effective, and commonly used machine learning classifier. Support Vector Machine is a training algorithm for learning classification and regression rules from data"

1.7 MODULE DESCRIPTION

- MODULE 1: Dataset collection
- MODULE 2: Data preprocessing
- MODULE 3: Exploratory Data Analysis
- MODULE 4: Model Fitting
- MODULE 5: Evaluation

MODULE 1: Dataset collection

Datasets are collected from Kaggle website. That dataset includes websites URL, that have both malicious and benign url.

There are 1781 rows × 21 columns datasets.

- ✓ Url_length
- ✓ Number_special_characters
- ✓ Charset
- ✓ Server
- ✓ Content_length
- ✓ Whois_country
- ✓ Whois_statepro
- ✓ Whois_regdate
- ✓ Whois_updated_date
- ✓ Tcp_conversation_exchange
- ✓ Dist_remote_tcp_port
- ✓ Remote_ips
- ✓ App_bytes

- ✓ Source_app_packets
- ✓ Remote_app_packets
- ✓ Source_app_bytes
- ✓ Remote_app_bytes
- ✓ App_packets
- ✓ Dns_query_times
- ✓ Type

MODULE 2: Data preprocessing

The sklearn.preprocessing package provides several common utility functions and transformer classes to change raw feature vectors into a representation that is more suitable for the downstream estimators.

In general, learning algorithms benefit from standardization of the data set. If some outliers are present in the set, robust scalers or transformers are more appropriate. The behaviors of the different scalers, transformers, and normalizers on a dataset containing marginal outliers is highlighted in Compare the effect of different scalers on data with outliers.

Standardization, or Mean removal and Variance Scaling

Standardization of datasets is a **common requirement for many machine learning estimators** implemented in scikit-learn; they might behave badly if the individual features do not more or less look like standard normally distributed data: Gaussian with **zero mean and unit variance.**

Scaling features to a range

In practice we often ignore the shape of the distribution and just transform the data to center it by removing the mean value of each feature, then scale it by dividing non-constant features by their standard deviation.

For instance, many elements used in the objective function of a learning algorithm (such as the RBF kernel of Support Vector Machines or the l1 and l2 regularizers of linear models) assume that all features are centered around zero and have variance in the same order. If a feature has a variance that is orders of magnitude larger than others, it might dominate the objective function and make the estimator unable to learn from other features correctly as expected.

An alternative standardization is scaling features to lie between a given minimum and maximum value, often between zero and one, or so that the maximum absolute value of each feature is scaled to unit size. This can be achieved using MinMaxScaler or MaxAbsScaler, respectively.

The motivation to use this scaling include robustness to very small standard deviations of features and preserving zero entries in sparse data.

MaxAbsScaler works in a very similar fashion, but scales in a way that the training data lies within the range [-1,1] by dividing through the largest maximum value in each feature. It is meant for data that is already centered at zero or sparse data.

Normalization

Normalization is the process of **scaling individual samples to have unit norm**. This process can be useful if you plan to use a quadratic form such as the dot-product or any other kernel to quantify the similarity of any pair of samples.

This assumption is the base of the Vector Space Model often used in text classification and clustering contexts.

Module 3: Exploratory Data Analysis

- Exploratory Data analysis (EDA) is used for visualize the datasets
- To visualize the dataset like pie chart, bar chart, box plot, histogram graph etc.,

Module 4: model fitting

In this proposed system we are using five machine learning algorithms named as Support Vector Machine (SVM), Logistic Regression, Random forest tree and naïve Bayes algorithms.

We could able to train the system using these four algorithms and evaluate training score calculated.

To classify the urls are Malicious or benign

Module 5: Evaluation

- The system will predict the url type with 98% accuracy
- To evaluate the accuracy score using confusion matrix method

A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known.

Confusion matrix include:

- Precision
- Recall
- Support
- F1score
- Accuracy

VI. CONCLUSION

Malicious web page identification is an emerging topic in cybersecurity. Though several research studies have been performed relating to the issues of malicious web page detection these are very costly as they consume more time and resources. In this research article, we employed a new web site classification system based on URL features to predict the web pages as malicious or benign using machine learning algorithms. The machine learning classifiers Random Forest (RF) achieves a higher accuracy of 95%. The experimental results have shown that our method can perform effectively for detecting the malicious web page.

REFERENCES

[1] Tao, Wang, Yu Shunzheng, and Xie Bailin. "A novel framework for learning to detect malicious web pages." In 2010 International Forum on Information Technology and Applications, vol. 2, pp. 353-357. Ieee, 2010.

[2] Eshete, Birhanu, Adolfo Villafiorita, and Komminist Weldemariam. "Malicious website detection: Effectiveness and efficiency issues." In 2011 First SysSec Workshop, pp. 123-126. IEEE, 2011.

[3] Aldwairi, Monther, and Rami Alsalman. "Malurls: A lightweight malicious website classification based on url features." Journal of Emerging Technologies in Web Intelligence 4, no. 2 (2012): 128-133.

[4] Yoo, Suyeon, Sehun Kim, Anil Choudhary, O. P. Roy, and T. Tuithung. "Two-phase malicious web page detection scheme using misuse and anomaly detection." International Journal of Reliable Information and Assurance 2, no. 1 (2014): 1-9.

[5] Hwang, Young Sup, Jin Baek Kwon, Jae Chan Moon, and Seong Je Cho. "Classifying malicious web pages by using an adaptive support vector machine." Journal of Information Processing Systems 9, no. 3 (2013): 395-404.

[6] Yue, Tao, Jianhua Sun, and Hao Chen. "Fine-grained mining and classification of malicious Web pages." In 2013 Fourth International Conference on Digital Manufacturing & Automation, pp. 616-619. IEEE, 2013.

[7] Krishnaveni, S., and K. Sathiyakumari. "SpiderNet: An interaction tool for predicting malicious web pages." In International Conference on Information Communication and Embedded Systems (ICICES2014), pp. 1-6. IEEE, 2014.

[8] Sun, Bo, Mitsuaki Akiyama, Takeshi Yagi, Mitsuhiro Hatada, and Tatsuya Mori. "Automating URL blacklist generation with similarity search approach." IEICE TRANSACTIONS on Information and Systems 99, no. 4 (2016): 873- 882.

Proceedings of the International Conference on Intelligent Computing and Control Systems (ICICCS 2020) IEEE Xplore Part Number:CFP20K74-ART; ISBN: 978-1-7281-4876-2.

[9] Urcuqui, Christian, Andres Navarro, Jose Osorio, and Melisa García. "Machine Learning Classifiers to Detect Malicious Websites." In SSN, pp. 14- 17. 2017.).

[10] Wang, Rong, Yan Zhu, Jiefan Tan, and Binbin Zhou. "Detection of malicious web pages based on hybrid analysis." Journal of Information Security and Applications 35 (2017): 68-74.74.

[11] Kim, Sungjin, Jinkook Kim, Seokwoo Nam, and Dohoon Kim. "WebMon: ML-and YARA-based malicious webpage detection." Computer Networks 137 (2018): 119-131.

[12] Altay, Betul, Tansel Dokeroglu, and Ahmet Cosar. "Context-sensitive and keyword density-based supervised machine learning techniques for malicious webpage detection." Soft Computing 23, no. 12 (2019): 4177-4191.