Network Security Project Report

By:
Muhmmad Maaz Siddiqui
Umar Khalid
Sara Ebrahim
Majeed Toorie

Friday 24th November, 2023

Contents

1 Task 1: Data Exploration and Cleaning

Explore the Kaggle dataset to gain insights into its structure, content, and the distribution of features. This dataset contains information about both malicious and benign webpages, encompassing crucial attributes like URL, HTML content, and additional features that will play a pivotal role in our analysis. Following the exploration, our next steps involve addressing data quality issues. This includes identifying and managing missing values, outliers, and inconsistencies within the dataset. Additionally, as part of the data preparation phase, we aim to standardize or normalize numerical features. This ensures that these features are on consistent scales, contributing to a more robust and reliable analysis.

2 Task 2: Feature Engineering

nhance feature richness by extracting additional information from various sources, including the URL, HTML content, and other pertinent data. Delve into the URL structure, dissect domain information, and scrutinize HTML tags. This process aims to derive features that may serve as indicators of malicious behavior. Subsequent to feature extraction, transform categorical features into numerical representations, a crucial step for compatibility with machine learning algorithms. To further refine our model and enhance efficiency, employ dimensionality reduction techniques. This strategy aims to streamline the dataset by reducing the number of features while optimizing overall model performance.

3 Task 3: Model Selection and Training

Conduct a comprehensive evaluation of diverse machine learning algorithms, encompassing logistic regression, support vector machines, Random Forest, and deep learning models. This evaluation specifically focuses on classifying webpages as either malicious or benign. Following algorithm selection, engage in hyperparameter tuning to fine-tune the performance of the chosen model. Subsequently, partition the dataset into training and testing sets for model training and evaluation. Proceed to train the chosen model on the training set, closely monitoring its performance to prevent overfitting and ensure its adaptability to new data.

4 Task 4: Model Evaluation and Validation

Assess the performance of the trained model on the testing set by computing essential metrics, including accuracy, precision, recall, and F1-score. These metrics provide a comprehensive understanding of the model's effectiveness in classifying webpages as malicious or benign.Analyze the model's performance on different types of malicious and benign websites. In addition to performance evaluation, we aim to identify any potential biases or limitations within

the model's predictions. This critical examination ensures a transparent understanding of the model's reliability and helps guide further refinement and adjustments.

5 Proposed Framework

1. Data Acquisition: Ensuring data quality and consistency.
2. Feature Engineering: Extract relevant features, encode categorical features.
3. Model Selection and Training: Evaluate and select appropriate machine learning algorithms, train the model on the dataset, and optimize hyperparameters.
4. Model Evaluation: Assess the model's performance on a testing set, evaluate metrics like accuracy, precision, recall, and F1-score.

5.1 Inclusions

- Data exploration and cleaning: This includes identifying and handling missing values, outliers, and inconsistencies in the dataset. It also includes standardizing or normalizing numerical features to ensure consistent scales.
- Feature engineering: This includes extracting additional features from the URL, HTML content, and other relevant sources. It also includes encoding categorical features into numerical representations.
- Model selection and training: This includes evaluating various machine learning algorithms for classifying webpages as malicious or benign. It also includes hyperparameter tuning to optimize the performance of the chosen model.
- Model evaluation and validation: This includes evaluating the trained model on a testing set, calculating metrics such as accuracy, precision, recall, and F1-score. It also includes analyzing the model's performance on different types of malicious and benign websites.
- Model documentation: This includes documenting the model's architecture, hyperparameters, and performance metrics. It also includes documenting any limitations or biases in the model.

5.2 Exclusions

- Data acquisition: This includes collecting and preparing the Kaggle dataset. It is assumed that the dataset is already available and ready for use.

- Model deployment: This includes integrating the trained model into a web application or API. It is assumed that the model will not be deployed in this phase.
- Model monitoring: This includes continuously monitoring the model's performance and retraining it with newer data. It is assumed that the model will not be monitored in this phase.

5.3 Hardware Requirements

- A computer with a processor that supports at least SSE4.2 instructions.
- 4GB of RAM.
- 10GB of free disk space.
- A graphics card with at least 2GB of memory.
- An internet connection.

5.4 Software Requirements

- The latest version of the Python programming language.
- The latest version of the all libraries used for Exploratory Data Analysis.

5.5 Vulnerability of Data Analysis Methodology

- The data analysis methodology used in this project is vulnerable to overfitting. This means that the model may perform well on the training data but poorly on new data.
- The data analysis methodology is also vulnerable to noise in the data. This means that the model may be sensitive to small changes in the data.

5.6 Functional Requirements

- The model should be able to classify websites as malicious or benign with an accuracy of at least 85%.
- The model should be able to classify websites with a precision of at least 80%.
- The model should be able to classify websites with a recall of at least 80%.