

Purpose of the Assignment

The goal of this task is to understand how computational intelligence approaches can assist in detecting vulnerabilities in web applications. Using real datasets (e.g., HTTP requests from CSIC 2010, CSE-CIC-IDS2018) or test cases from the Open Web Application Security Project (OWASP), you will build a model that identifies vulnerable requests (e.g., SQL Injection, XSS).

Algorithm Selection

Choose one algorithm from the supervised learning group (a branch of machine learning):

- **Logistic Regression**
 - Used for binary classification (e.g., yes/no, 0/1).
 - Predicts the probability that an instance belongs to a certain class.
- **Random Forest**
 - Composed of multiple decision trees that vote together.
 - Used for classification and regression.
 - Robust and often highly accurate.
- **K-Nearest Neighbors (KNN)**
 - Classifies an instance based on the majority class among its K nearest neighbors.
 - Simple but can be very effective for classification.
- **Your own choice of classification algorithm.**

Note: You may use already implemented algorithms from libraries (e.g., Weka - <https://ml.cms.waikato.ac.nz/weka>).

Task Description

Part 1: Data Preparation

Use a publicly available dataset, such as:

- **CSIC 2010 HTTP dataset** (benign vs. attack):
<https://www.kaggle.com/datasets/ispangler/csic-2010-web-application-attacks>
- **CSE-CIC-IDS2018** (includes web attacks):
<https://www.kaggle.com/datasets/dhoogla/csecicids2018>
- Specialized datasets for SQL Injection or Cross-Site Scripting (XSS).

Sample size: at least **10,000 requests**. Split: **training/testing** (e.g., 80/20).

Part 2: Analysis and Feature Extraction

Compute basic statistics (e.g., attack ratio, request length).
Extract important features such as:

- URL and parameter length
 - Number of special characters
 - Entropy
 - TF-IDF n-grams
-

Part 3: Modeling

Train the selected model on the data. Optimize hyperparameters (e.g., prevent overfitting using regularization, number of trees, K).

Part 4: Evaluation

Analyze False Positives (FP) and False Negatives (FN).

Metrics: Accuracy, F1, ROC-AUC.

Visualize results (e.g., confusion matrix, histogram score).

Part 5: Report

Include:

- Data and feature description
- Selected algorithm and parameters
- Results with graphs
- Interpretation (e.g., for important features)
- Limitations and recommendations

Submission: Archive (zip/rar/7z – max 50 MB) containing:

- Code
 - Data sample (up to 10k rows)
 - Report (PDF) with analysis, results, and conclusions.
-

Brief explanation of basic statistics and feature extraction

Basic Statistics

- Attack ratio: proportion of attack vs. benign requests.
- Request length: average, minimum, and maximum length of HTTP requests (URL + parameters).
- Purpose: check data structure and potential imbalance.

Important Features

- URL and parameter length: attacks are often longer.
- Number of special characters: e.g., ' , " , < , > – typical for SQL Injection and XSS.
- Entropy: measures randomness; high entropy may indicate obfuscation.
- TF-IDF n-grams: textual representation of URLs and parameters; n-grams help detect attack patterns (e.g., UNION SELECT, <script>).

Why is this important? Raw data (HTTP requests) can be transformed into numerical features that the model uses to classify attacks.