MULTILABEL CAPEC CLASSIFICATION OF WEB ATTACKS

Enrollment Numbers: 21103218, 21104032, 21104023

Name : Aibad Khan, Akshat Agrawal, Pritpal Singh

Name of Supervisor: Dr. Sangeeta Mittal

November - 2023

Submitted in Partial Fulfillment of Degree of
Bachelor of Technology in
Information Technology

INTRODUCTION

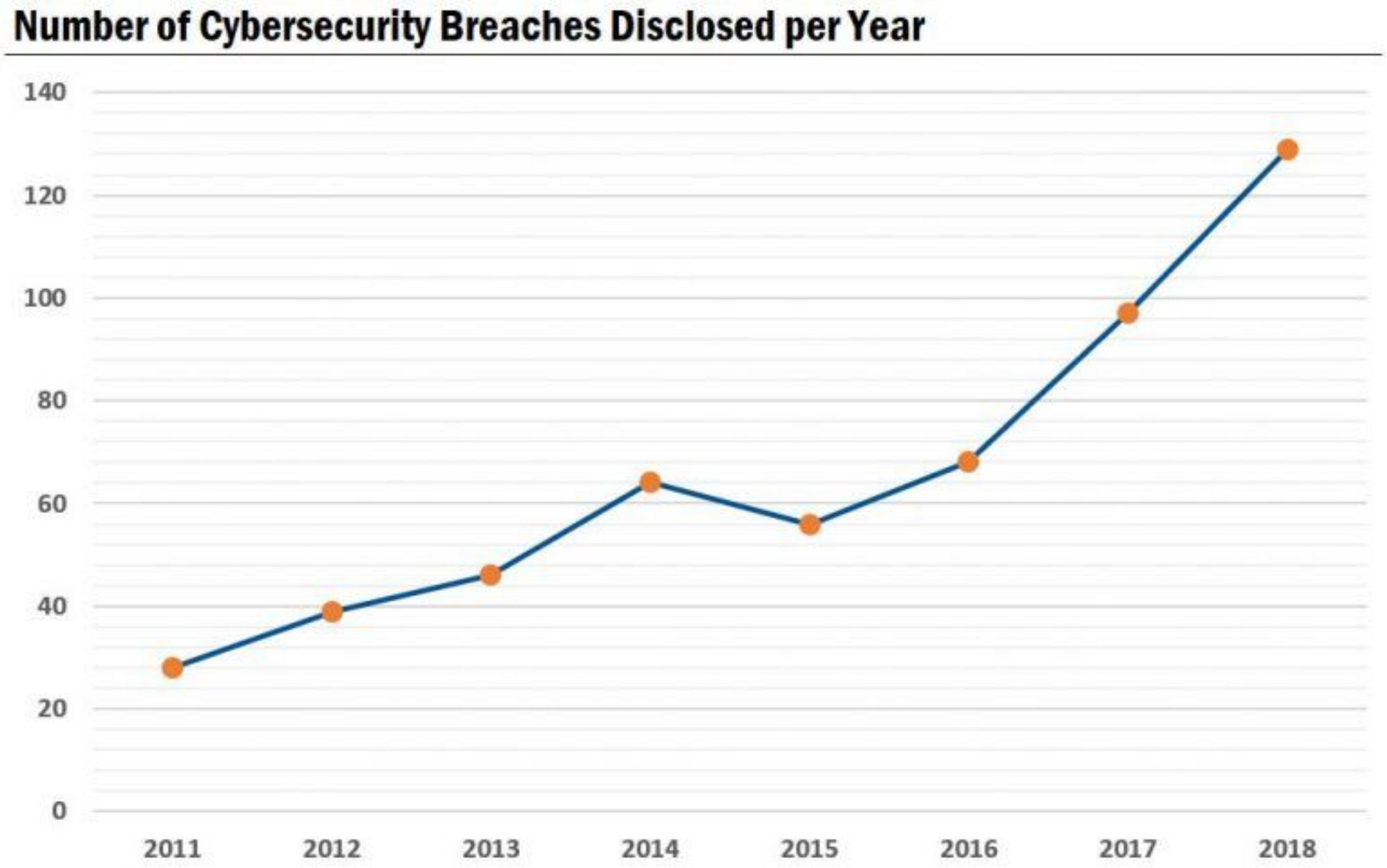


Fig1. Increase in The Number of Internet Users According to Years (2011- 2018)

Source: https://blog.auditanalytics.com/trends-in-cybersecurity-breaches-continue-in-2019/ as seen on 20 Nov 2023

Inspiration & Motivation

Escalating Cyber Threats: The Alarming Surge in Attacks on Web Servers, E-commerce Platforms, and Critical Institutions. Every year there is a significant increase in the number of attacks against web servers and applications, e-commerce platforms, financial and government institutions, large corporations, etc. are targeted by web attacks for economic or ideological reasons.

PROBLEM STATEMENT

To build a model for identifying the web attacks and classify them under the CAPEC catalog by applying MultiLabel Classification techniques

OBJECTIVE

3.1 WHAT ARE WEB ATTACKS ? 🤔

Web attacks are malicious activities aimed at exploiting vulnerabilities or weaknesses in websites, web applications, or web services.

3.2 TYPES OF WEB ATTACKS

- **Cross-Site Scripting (XSS)**: Injecting malicious scripts into web pages viewed by other users.
- **SQL Injection**: Inserting SQL code into input fields to manipulate or retrieve sensitive data from a database.
- **Cross-Site Request Forgery (CSRF)**: Tricking users into executing unwanted actions on a web application where they are authenticated.
- **Phishing**: Deceptive techniques to acquire sensitive information like login credentials, often through fake websites or emails.
- **DDoS (Distributed Denial of Service)**: Overwhelming a web server with a flood of traffic, causing it to become unavailable to legitimate users.

3.3 CAPEC CLASSIFICATION

CAPEC stands for Common Attack Pattern Enumeration and Classification. It is a structured list of common attack patterns used by adversaries to exploit vulnerabilities in software and systems.

Each entry in the CAPEC list includes:

- ➢ **Attack Pattern Name**: A descriptive name(eg.path traversal,sql injection) for the attack method.
- ➢ **Description**: Details about the attack pattern, including its goals, tactics, and techniques used by attackers.
- ➢ **Likelihood of Attack**: An assessment of how probable the attack is to occur.
- ➢ **Methods of Attack**: Specific steps or procedures utilized by attackers within the pattern.
- ➢ **Examples**: Instances or scenarios where this attack pattern has been observed.

ABOUT DATASET

- Most datasets are composed of artificially generated traffic and, to our best knowledge, all datasets available for training and/or evaluation of machine learning models provide only labeling of the request in terms of normality or attack, without specifying in any case what type(s) of attack(s) is/are being suffered.

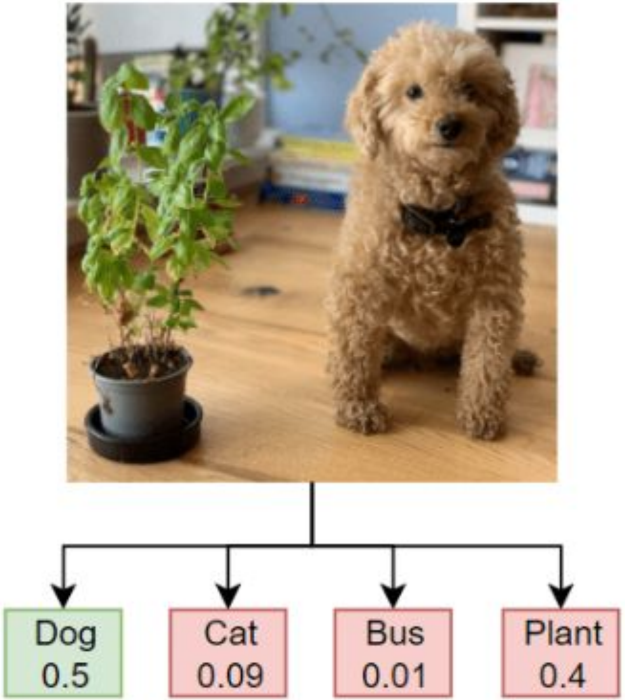
- For this reason, one of the main achievements of this work is the generation of a new dataset (SR-BH 2020 dataset) that collects different types of attacks, coming from real traffic data (generated by collecting real traffic in a honeypot exposed to the Internet for 12 days), with multi-labels, that report the normality of the request, or the CAPEC classification of the type or types of attack that the web request represents.

- This dataset, to our knowledge, is the first one that allows the training and evaluation of multi-label machine learning models and algorithms, which can provide the CAPEC classification of the attack(s) that a web application is suffering.

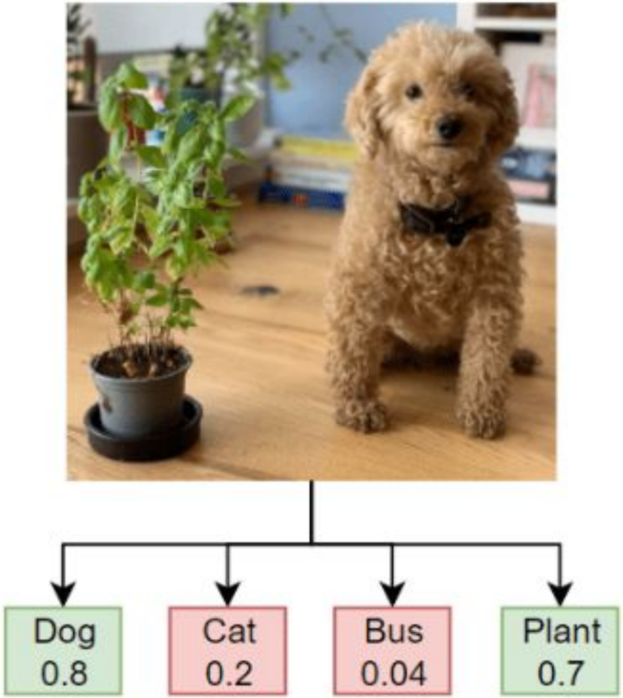
MULTILABEL VS MULTICLASS CLASSIFICATION

Example

Multiclass Classification



Multilabel Classification



- ➢ In multiclass classification, the problem involves assigning an input to one of several classes or categories. Each input belongs to exactly one class out of multiple possible classes.

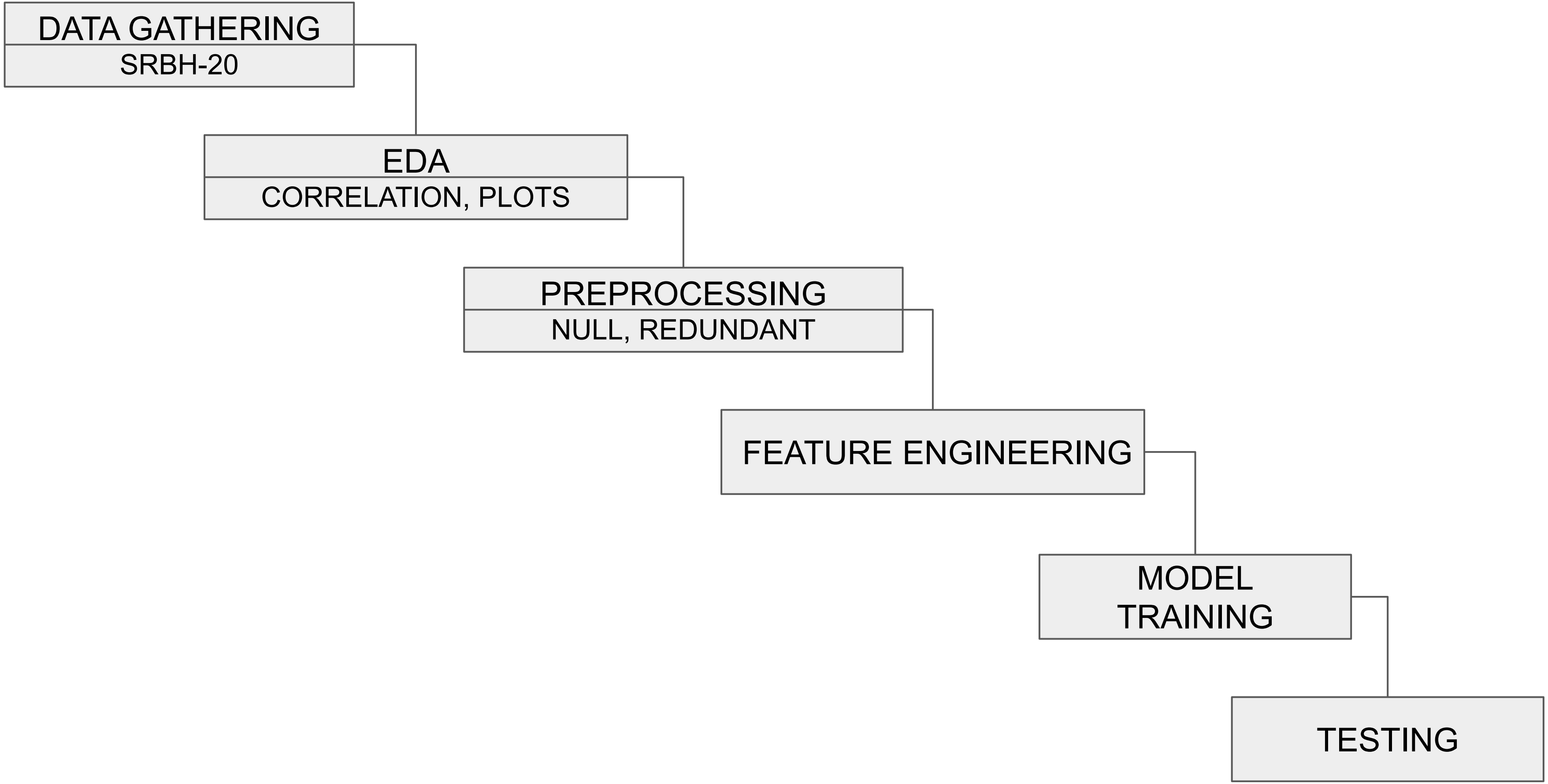
- ➢ Multilabel classification, on the other hand, deals with assigning multiple labels or categories to each input. This means that an input can belong to more than one category simultaneously.

ALGORITHMS FOR MULTILABEL CLASSIFICATION

There are several algorithms used for multilabel classification, each with its strengths and suitability for different types of data and problem scenarios. Some of the commonly used algorithms for multilabel classification include:

- ➢ **Binary Relevance**: Treats each label as a separate binary classification problem.
- ➢ **Classifier Chains**: Chains binary classifiers, using previous predictions as features for subsequent labels.
- ➢ **Label Powerset**: Converts multilabel problem into a multi-class problem, considering each label combination as a distinct class.
- ➢ **Adapted Algorithm**: Traditional algorithms modified or extended for multilabel classification.
- ➢ **Deep Learning Models**: Neural network architectures for complex data like images, sequences, or text.
- ➢ **Ensemble Methods**: Combine outputs of multiple classifiers for improved performance.

STEPS OF EXECUTION



DATA GATHERING

SRBH-20

EDA

CORRELATION, PLOTS

PREPROCESSING

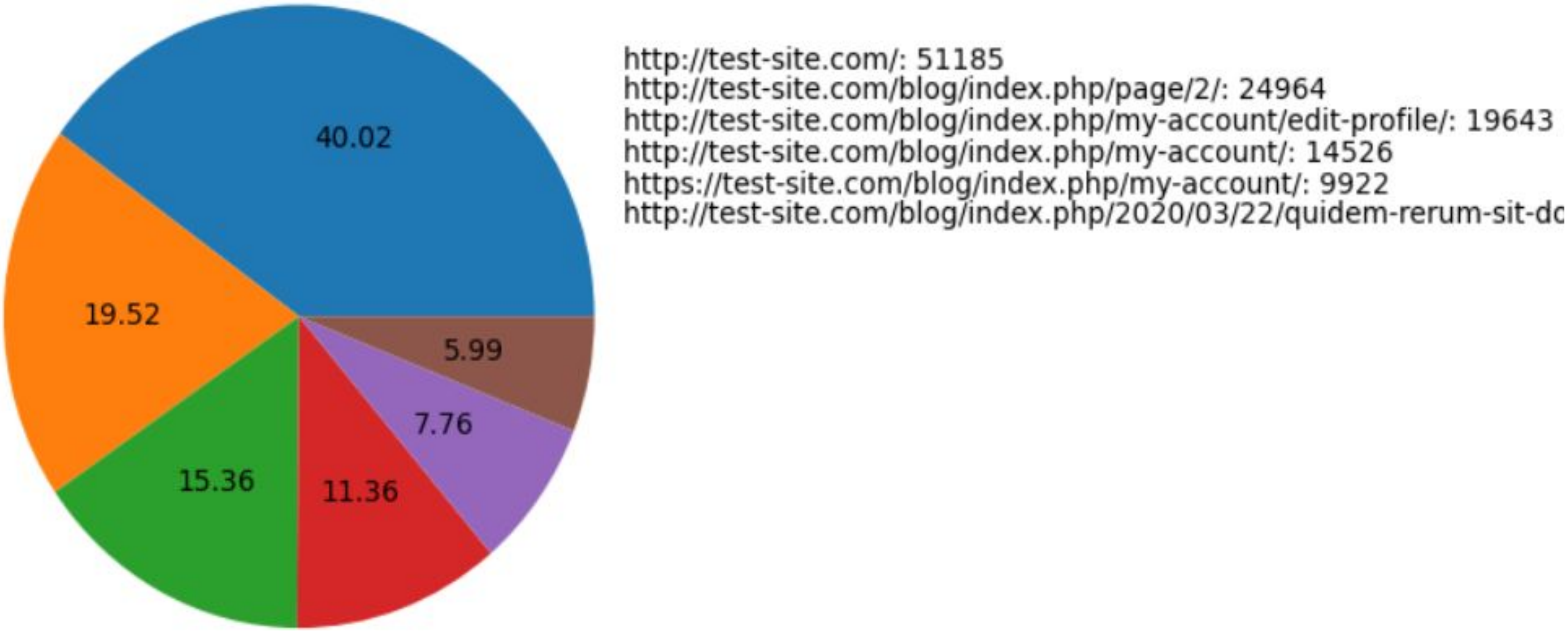
NULL, REDUNDANT

FEATURE ENGINEERING

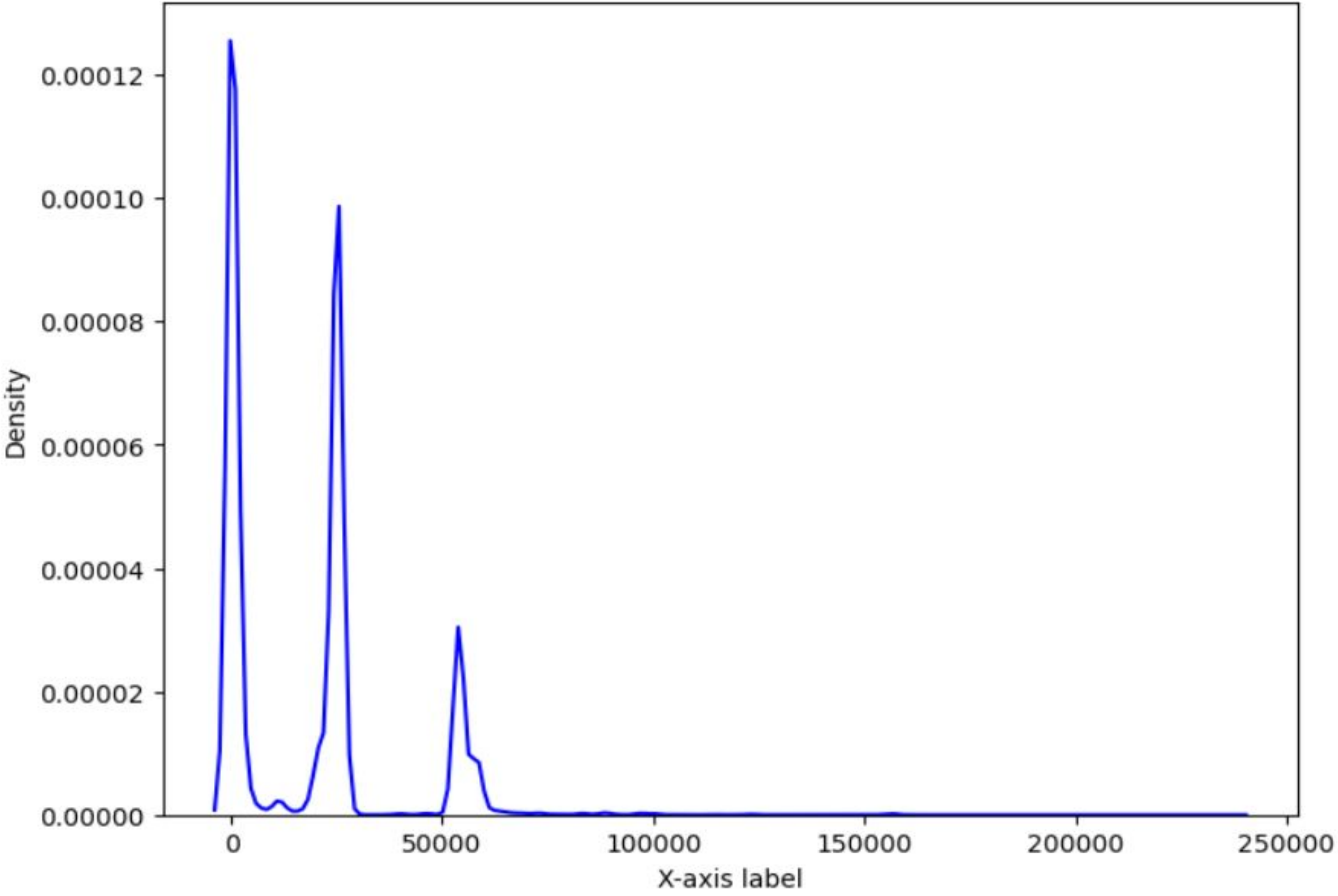
MODEL TRAINING

TESTING

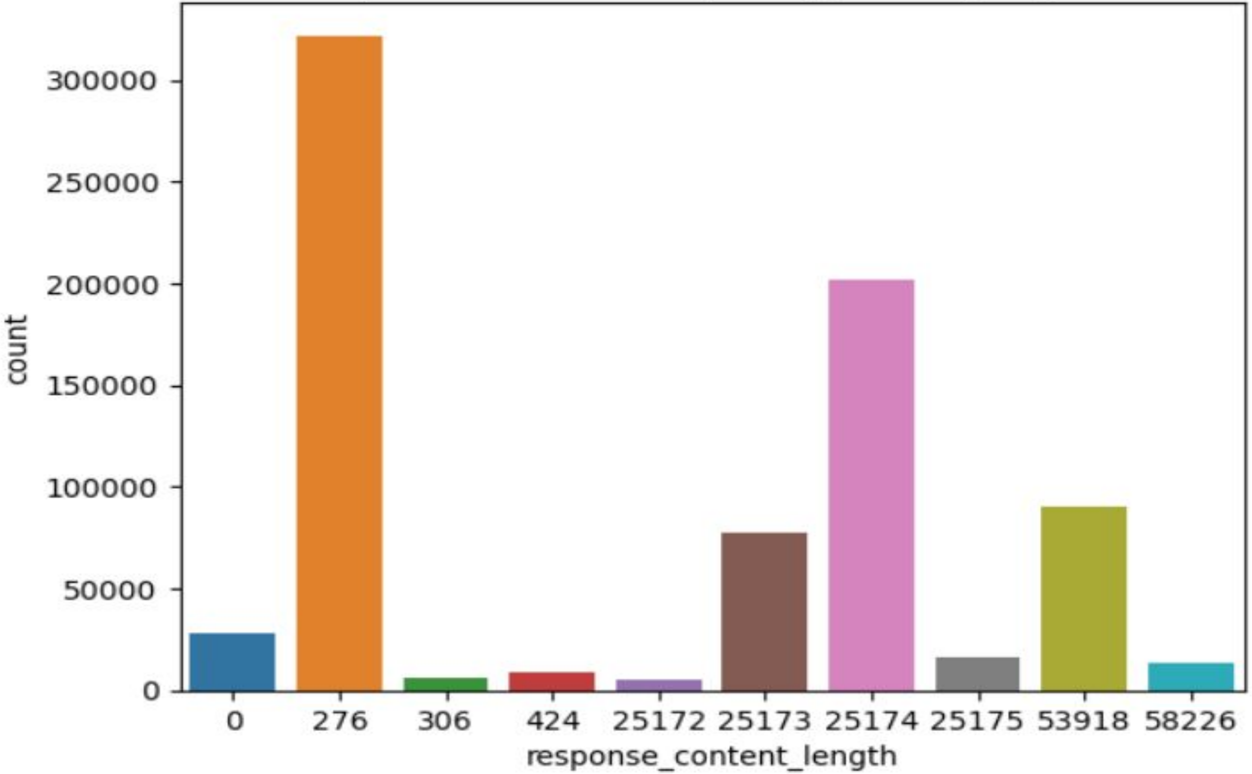
EXPLORATORY DATA ANALYSIS



request_referer column’s distribution of values



Kernel Density Estimate(KDE) Plot of Numerical Data



Top 10 counts of response_content_length

DATA PREPROCESSING

- Handle missing values: Removed missing data points based on the nature and extent of missingness.

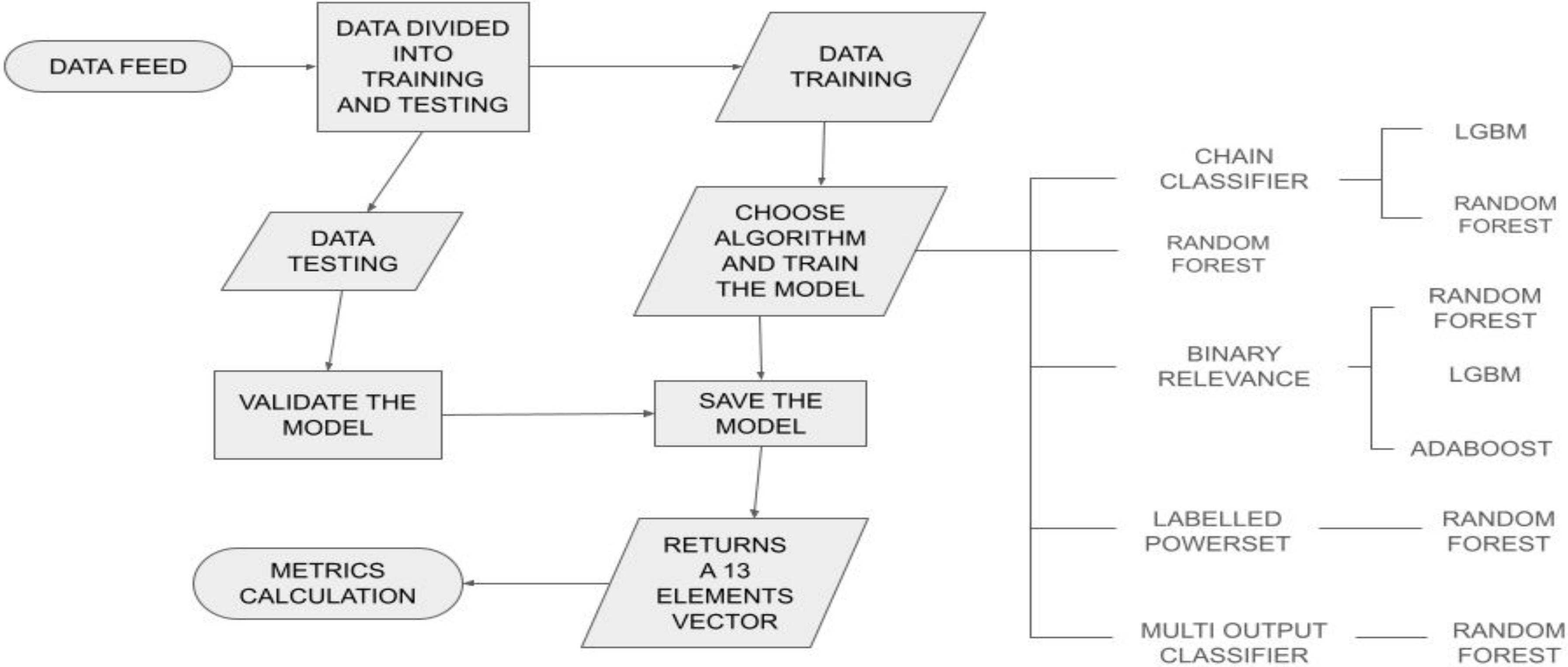
- Removed duplicate values from the dataset

- Convert categorical labels (e.g., 'normal,' 'protocol manipulation') into a format that machine learning models can understand(Count Encoding in our case).

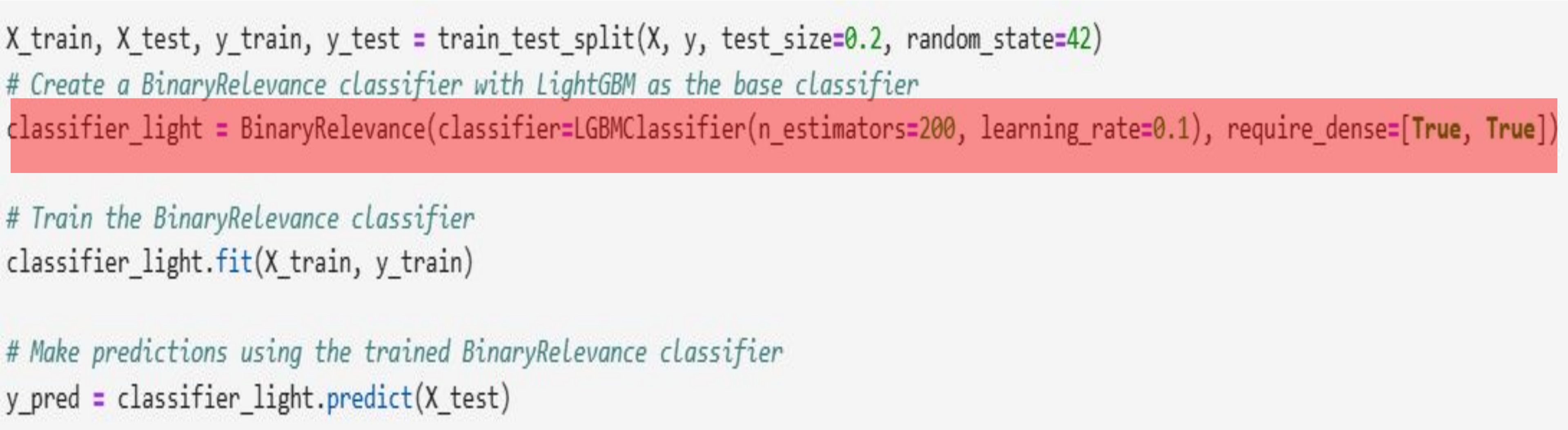
- Split the dataset into training and testing sets to evaluate model performance accurately.

IMPLEMENTATION

Flow Chart



DATA FEED
DATA DIVIDED INTO TRAINING AND TESTING
DATA TRAINING
DATA TESTING
CHOOSE ALGORITHM AND TRAIN THE MODEL
VALIDATE THE MODEL
SAVE THE MODEL
RETURNS A 13 ELEMENTS VECTOR
METRICS CALCULATION
CHAIN CLASSIFIER
LGBM
RANDOM FOREST
RANDOM FOREST
BINARY RELEVANCE
RANDOM FOREST
LGBM
ADABOOST
LABELLED POWERSET
RANDOM FOREST
MULTI OUTPUT CLASSIFIER
RANDOM FOREST

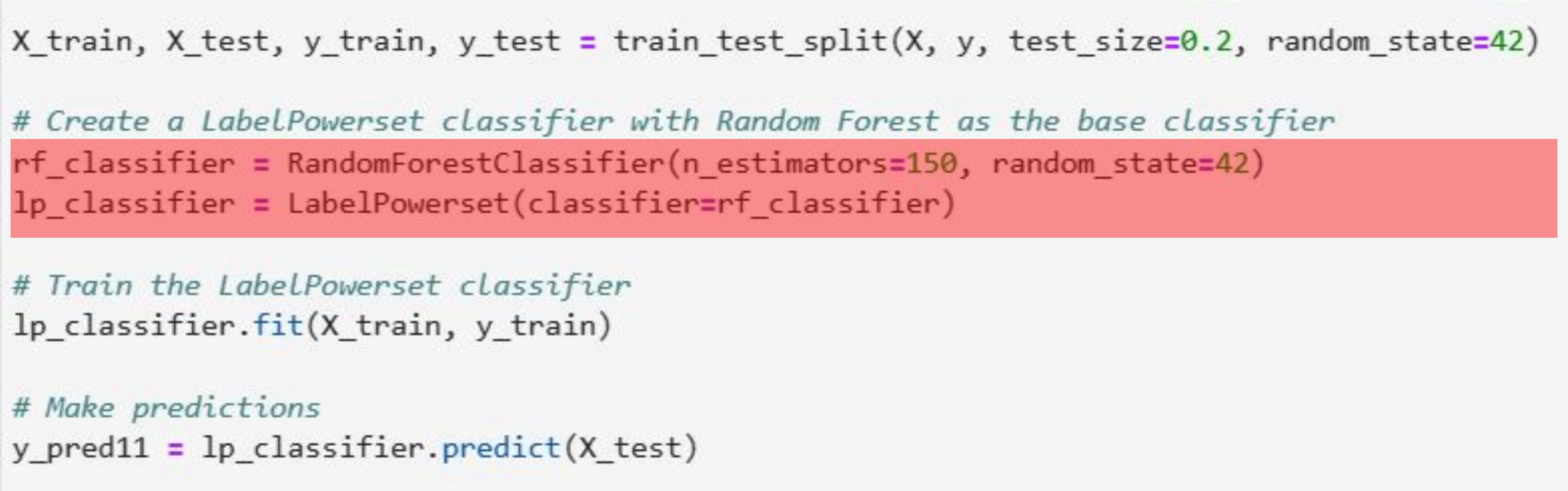


```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
# Create a BinaryRelevance classifier with LightGBM as the base classifier
classifier_light = BinaryRelevance(classifier=LGBMClassifier(n_estimators=200, learning_rate=0.1), require_dense=[True, True])

# Train the BinaryRelevance classifier
classifier_light.fit(X_train, y_train)


# Make predictions using the trained BinaryRelevance classifier
y_pred = classifier_light.predict(X_test)
```

Binary Relevance+LGBM



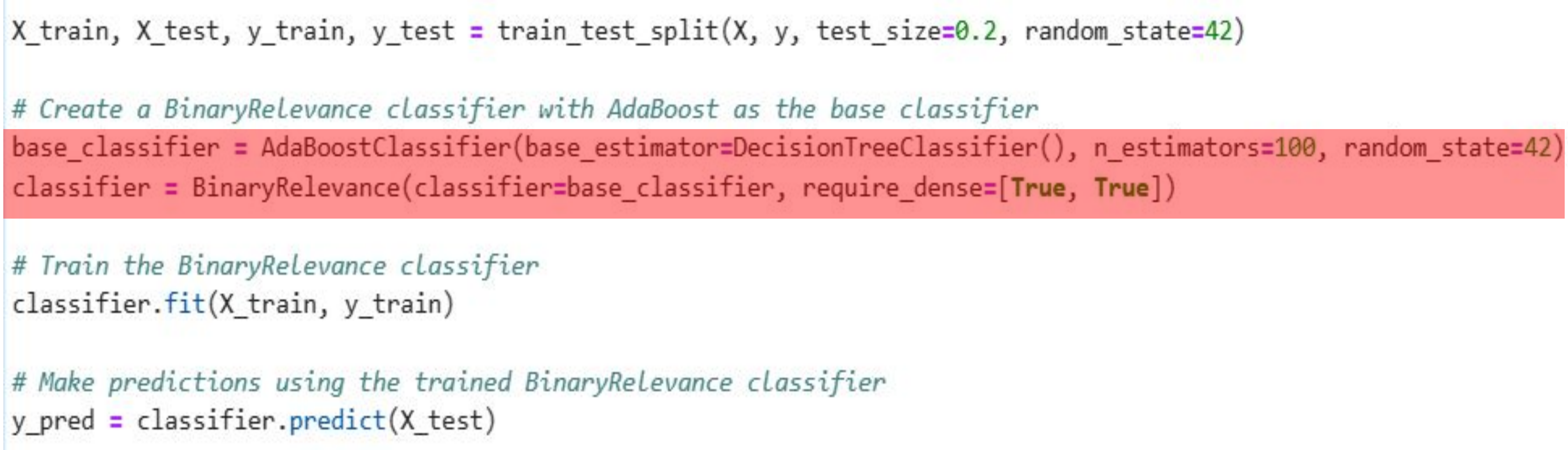
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Create a LabelPowerset classifier with Random Forest as the base classifier
rf_classifier = RandomForestClassifier(n_estimators=150, random_state=42)
lp_classifier = LabelPowerset(classifier=rf_classifier)

# Train the LabelPowerset classifier
lp_classifier.fit(X_train, y_train)

# Make predictions
y_pred11 = lp_classifier.predict(X_test)
```

Label Powerset+Random Forest



```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Create a BinaryRelevance classifier with AdaBoost as the base classifier
base_classifier = AdaBoostClassifier(base_estimator=DecisionTreeClassifier(), n_estimators=100, random_state=42)
classifier = BinaryRelevance(classifier=base_classifier, require_dense=[True, True])

# Train the BinaryRelevance classifier
classifier.fit(X_train, y_train)

# Make predictions using the trained BinaryRelevance classifier
y_pred = classifier.predict(X_test)
```

Binary Relevance+Adaboost

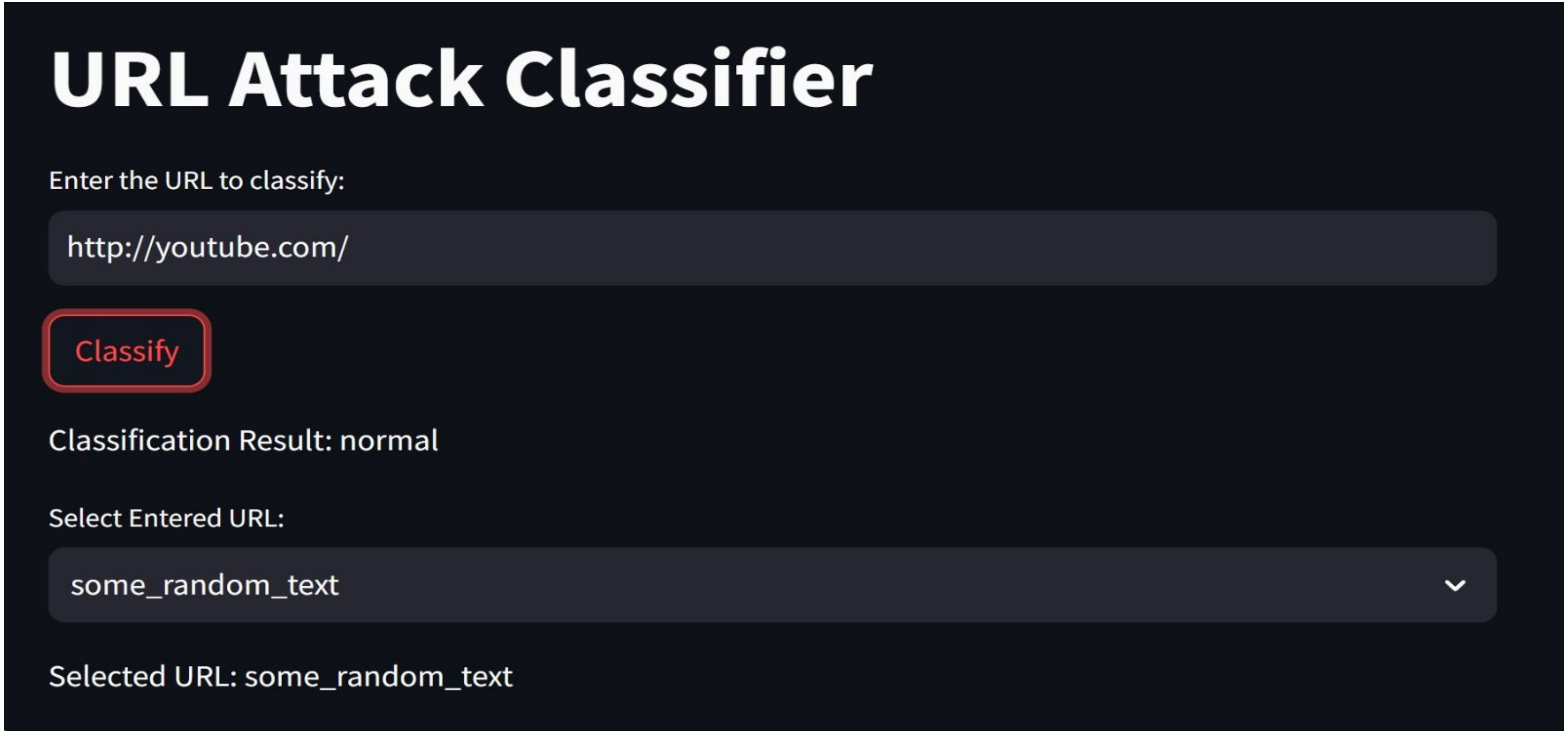
RESULTS

- The results of the algorithms and model combinations have been evaluated according to the metrics and scenario and a brief detail is attested in the next slide.

Table 6.

Summary of metrics by algorithm and model, ordered by recall score.

Model	Accuracy	F1-Micro	Hamming Loss	Precision Micro	Recall	ROC AUC	Jacquard Sim.	Informed.	Marked.
Binary Relevance + Random forest	0.81246	0.85553	0.02135	0.90286	0.81292	0.90277	0.74754	0.74953	0.77892
Binary Relevance + AdaBoost	0.80549	0.85353	0.02154	0.90572	0.80703	0.89997	0.74449	0.74277	0.77886
Binary Relevance + LightGBM	0.80321	0.85442	0.02135	0.90981	0.80539	0.89932	0.74584	0.75663	0.78360
Classifier Chain + Random Forest	0.85490	0.85603	0.02236	0.85728	0.85479	0.92139	0.74830	0.80709	0.72951
Labeled Power Set + Random Forest	0.85450	0.85571	0.02242	0.85681	0.85462	0.92128	0.74781	0.80572	0.73010
Multi o/p Classifier + Random Forest	0.81246	0.85553	0.02135	0.90286	0.81292	0.92277	0.74754	0.78643	0.76010
Random Forest	0.81285	0.85552	0.02136	0.90279	0.81295	0.90278	0.74751	0.74953	0.77892



URL Attack Classifier

Enter the URL to classify:

http://youtube.com/

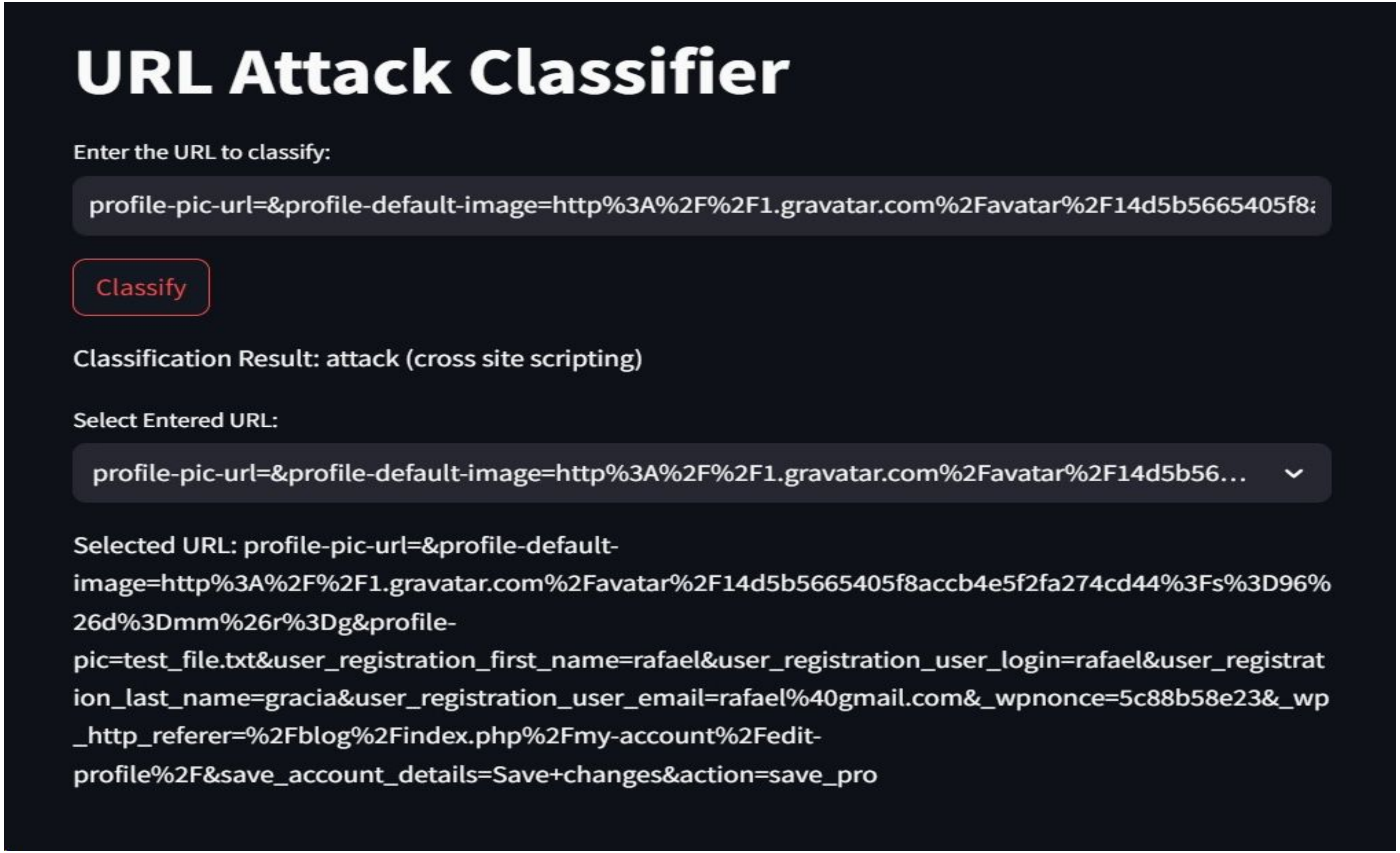
Classification Result: normal

Select Entered URL:

some_random_text

Selected URL: some_random_text

Case when the input url is ‘normal’ (not an attack)



URL Attack Classifier

Enter the URL to classify:

profile-pic-url=&profile-default-image=http%3A%2F%2F1.gravatar.com%2Favatar%2F14d5b5665405f8a

Classification Result: attack (cross site scripting)

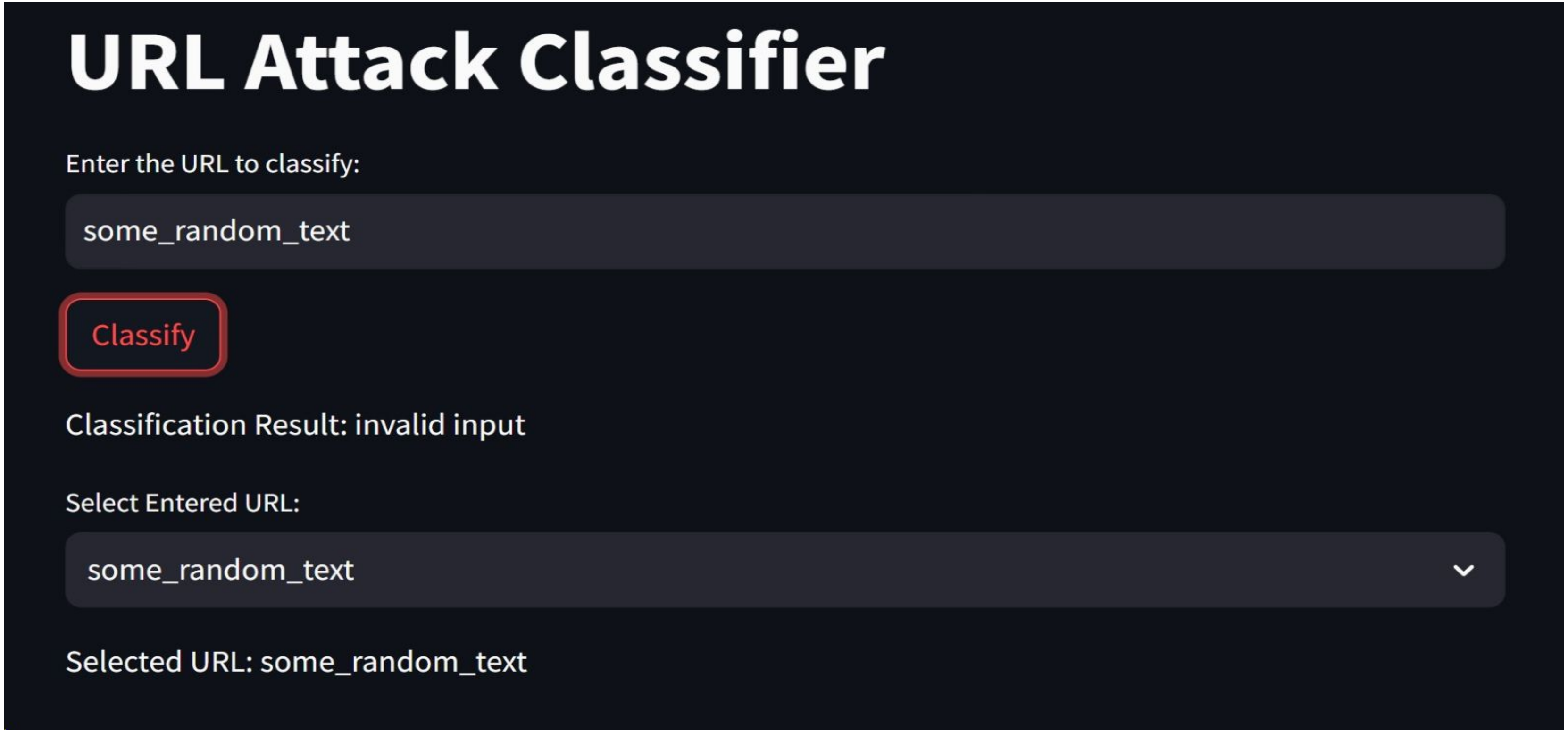
Select Entered URL:

profile-pic-url=&profile-default-image=http%3A%2F%2F1.gravatar.com%2Favatar%2F14d5b56...

Selected URL: profile-pic-url=&profile-default-image=http%3A%2F%2F1.gravatar.com%2Favatar%2F14d5b5665405f8accb4e5f2fa274cd44%3Fs%3D96%26d%3Dmm%26r%3Dg&profile-pic=test_file.txt&user_registration_first_name=rafael&user_registration_user_login=rafael&user_registration_last_name=gracia&user_registration_user_email=rafael%40gmail.com&_wpnonce=5c88b58e23&_wp_http_referer=%2Fblog%2Findex.php%2Fmy-account%2Fedit-profile%2F&save_account_details=Save+changes&action=save_pro

Case when the input url is ‘normal’ (not an attack)

Testing different cases for frontend



URL Attack Classifier

Enter the URL to classify:

some_random_text

Classification Result: invalid input

Select Entered URL:

some_random_text

Selected URL: some_random_text

Case when the input is invalid(not a url)

CONCLUSION

- ➢ In this work we have presented the SR-BH 2020 multi-label dataset, which includes a set of 13 different labels, providing information about the normality of each web request and its possible classification into 12 different CAPEC categories.

- ➢ We have also designed and evaluated different multi-label classification models, using modules and classes from the scikit learn and scikit-multi learn libraries.

- ➢ Two leading algorithms in the field of machine learning have been tested with these models: LightGBM and AdaBoost.

- ➢ The results obtained by our experiments show a clear superiority of the combination of the CatBoost algorithm and the two-phase model with the MultiOutputClassifier module of the scikit-learn library, in multi-label classification tasks.

FUTURE WORK

Cross-Validation Strategies:

- Experiment with different cross-validation strategies to ensure robustness of your model evaluations.

Hyperparameter Tuning:

- Conduct a more extensive hyperparameter tuning process for the selected algorithms. This can involve using techniques like grid search, random search, or more advanced optimization algorithms to fine-tune model parameters and improve overall performance.

Deep Learning Approaches:

- Deep learning architectures may uncover intricate patterns in the data that traditional machine learning algorithms might miss.

Model Deployment and Monitoring:

- If applicable, explore the challenges and considerations for deploying your model in a production environment. Consider implementing monitoring mechanisms to track model performance over time and ensure its continued effectiveness.

External Datasets and Transfer Learning:

- Explore the possibility of incorporating external datasets that might provide additional relevant information for your classification task. Transfer learning techniques could be considered if pre-trained models on related tasks are available.

THANK YOU!