**Abstract**

The growing reliance on web-based applications has necessitated robust and secure systems capable of safeguarding sensitive information. However, the complexity of these applications, particularly their use of JavaScript methods and properties, often leaves them vulnerable to exploitation. To address this, we propose a machine learning-based approach that detects malicious web pages based on their usage of JavaScript methods and properties.

Our approach leverages a dataset extracted from various websites, detailing their usage of different JavaScript methods and properties. We employ a combination of Random Forest and XGBoost classifiers, tuned using GridSearchCV, to predict whether a website is malicious based on this data. This novel approach provides a real-time solution for detecting malicious web pages, offering a significant contribution to the field of cybersecurity.

The expected outcome of this research is a functional machine learning model capable of accurately predicting malicious web pages. This not only enhances the security of web-based applications but also provides valuable insights into the usage patterns of JavaScript methods and properties across different websites.

**Introduction**  
  
As we continue to delve deeper into the era of digitization, our reliance on the internet has become an inseparable facet of modern life. We access this vast cyberspace primarily through web browsers, a convenience that has its darker sides. This easy access has, unfortunately, opened the floodgates for cybercriminals who exploit the web to conduct illicit activities by crafting malevolent web pages. As highlighted by Kaspersky's 2021 report, around 25% of users encountered at least one web-based attack [^1^]. Concurrently, phishing, and crypto-related offenses were identified as the leading types of cyber threats, accounting for 85% of all identified threats that year, as reported by Cisco [^2^].

These rising instances of cyber-attacks underline the imperative need for innovative and effective methodologies to detect and neutralize malevolent web pages. Traditional detection techniques, including static and dynamic analysis, each present unique advantages but also pose considerable constraints. Static analysis, albeit simple, grapples with high false negative rates. On the other hand, dynamic analysis, despite providing comprehensive insights, is resource intensive. Moreover, existing machine learning and deep learning-based detection strategies frequently depend on single-modal features or elementary stacking of classifiers, thereby risking information loss and compromising the accuracy of the overall methodology.

Given these constraints, our study introduces a sophisticated multi-modal learning-based model to detect malicious web pages. This model employs machine learning algorithms, specifically Random Forest and XGBoost classifiers, and optimizes their performance using hyperparameter tuning through GridSearchCV. The model's robustness is further enhanced by leveraging an ensemble learning approach, which amalgamates predictions from both classifiers to deliver more accurate detection. Importantly, this method is not contingent on web content, such as text, and is therefore proficient in detecting malevolent web pages in multiple languages, thus broadening its universal relevance.

The proposed novel model presents encouraging outcomes, indicating its efficacy in mitigating the continually expanding threat of web-based attacks. By perpetually enhancing our detection methodologies, we aspire to maintain the cyberspace as a secure environment for all its users.

**Related work**

The field of cybersecurity has seen a significant increase in research related to the detection of malicious web pages. Various machine learning models have been proposed and implemented to tackle this problem. In this section, we will discuss four such research papers that are closely related to our work.

The first paper, titled "Detection and Analysis of Drive-by Download Attacks and Malicious JavaScript Code" [1], presents a comprehensive analysis of drive-by download attacks, a common method used by attackers to spread malware. The authors propose a method for detecting malicious JavaScript code, which is a key component of these attacks. This is particularly relevant to our work as our dataset includes features related to JavaScript methods and properties used by websites. However, our approach differs in that we use a machine learning model to predict the maliciousness of a website based on these features, rather than focusing solely on JavaScript code analysis.

The second paper, "MM-ConvBERT-LMS" [2], introduces a novel machine learning model for text classification tasks. Although the model was not specifically designed for malicious URL detection, its architecture and techniques can be adapted for our purposes. our approach focuses on using ensemble machine learning model

The third paper, "Detection of Malicious Web Pages Based on Hybrid Analysis" [3], proposes a hybrid analysis method for detecting malicious web pages. The authors combine static and dynamic analysis to extract features from web pages, which are then used to train a machine learning model. This approach is similar to ours in that we also extract features from web pages (i.e., usage of JavaScript methods and properties) to train our model. However, our work differs in the specific features used and the machine learning models employed.

The fourth paper, "A Novel Approach for Malicious URL Detection Based on the Joint Model" [4], presents a joint model for malicious URL detection that combines lexical and host-based features. This approach is relevant to our work as it also involves predicting the maliciousness of a web page. However, our work differs in the features used (JavaScript methods and properties vs. lexical and host-based features) and the machine learning model employed.

**Method**

Our proposed method for detecting malicious web pages involves using an ensemble machine learning model that combines the strengths of the Random Forest and XGBoost models. The ensemble model is trained on a dataset that includes various JavaScript methods and properties used by websites, as well as a binary target variable indicating whether each website is benign or malicious.

The Random Forest model is a powerful and versatile machine learning model that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes of the individual trees. For classification, two common criteria are used:

Gini Impurity:   
This is a measure of how often a randomly chosen element from the set would be incorrectly labeled if it was randomly labeled according to the distribution of labels in the subset. It is calculated as:

Gini Impurity = 1 - ∑ (p\_i)^2

where p\_i is the probability of an element being classified to a particular class.

Entropy:   
This is a measure of impurity, disorder, or uncertainty. The entropy of a set is zero when it contains instances of only one class. The entropy is calculated as:

Entropy = - ∑ p\_i \* log2(p\_i)

where p\_i is the probability of an element being classified to a particular class.

The XGBoost model, on the other hand, is an implementation of gradient boosted decision trees designed for speed and performance. Gradient boosting is a method that goes through cycles to iteratively add models into an ensemble. It begins by initializing the ensemble with a single model, whose predictions can be naive. Then, we start the cycle:

First, we use the current ensemble to generate predictions for each observation in the dataset. To make a prediction, we add the predictions from all models in the ensemble. These predictions are used to calculate a loss function (like mean squared error, for instance).Then, we use the loss function to fit a new model that will be added to the ensemble. Specifically, we determine model parameters so that adding this new model to the ensemble will reduce the loss.

The objective function to be optimized is given by:

Obj(Θ) = Σi=1n l(yi, y^i) + Σj=1m Ω(fj)

where l is a differentiable convex loss function that measures the difference between the prediction y^i and the target yi, and Ω is a regularization term that penalizes the complexity of the model.

The prediction of the XGBoost model is given by a weighted sum of K base learners (decision trees):

y^i= Σk=1K fk(xi)

where fk is a function in the functional space F, representing the structure of the decision tree.

The ensemble model combines the predictions of the Random Forest and XGBoost models using a voting mechanism.

E(x) = argmaxv Σj=1J I(vj = v)

where E(x) is the prediction of the ensemble model, vj is the prediction of the j-th model, and I is the indicator function that equals 1 if vj = v and 0 otherwise.

**Experiment**

**Dataset**List of the most popular websites were collected from <https://tranco-list.eu/> and the list of malicious websites were collected from <https://urlhaus.abuse.ch/> for this experiment.

A web-addon was built for this experimentation. The task of this addon was to collect the count browser APIs that the browser called when it visited a list of websites. This is an automated browser that installs an add-on as soon as it is initiated. The browser add-on then crawls through the list of websites provided to it. Two different instances are being used over here to collect data. One for malicious websites and other for benign websites. The data collected from these automated browsers are stores in excel files. The data transformation step after the collection of data involves, flagging of malicious and benign websites. This excel sheet is then concatenated and converted to excel files.

After the data is transformed, it is loaded to the machine learning model, which in turn trains the model and then tests the model for accuracy and other evaluation metrics.

**Conclusion**

In this paper, we proposed a method for detecting malicious web pages using an ensemble machine learning model that combines the strengths of the Random Forest and XGBoost models. Our model was trained on a dataset that includes various JavaScript methods and properties used by websites, with a binary target variable indicating whether each website is benign or malicious.

Our experiment showed that our ensemble model performed well in correctly identifying both benign and malicious websites, achieving an accuracy of 0.873, a precision of 0.86 for benign websites and 0.89 for malicious websites, a recall of 0.90 for benign websites and 0.85 for malicious websites, and an F1-score of 0.88 for benign websites and 0.87 for malicious websites.