# Abstract

The study involves state-of-the-art machine learning approaches to detect Cross-Site Scripting attacks, which is one of the most severe types of threats to web application security. In this study, three machine learning models, namely Random Forest, Gradient Boosting, and XGBoost, have been implemented and evaluated using an extensive dataset of over 8.5 million labeled samples of network traffic which were reduced to 100000 samples. The focus of the analysis is to identify network traffic by classifying it as either benign or malicious, based on the most appropriate key features such as MIN\_IP\_PKT\_LEN and PROTOCOL. XGBoost is the most robust among the models presented in this paper, with accuracy after tuning at 99.74%, outperforming all other methods. Model results were evaluated using accuracy, precision, recall, and F1-score while cross-validation demonstrated consistency across the models. Although the study confirms the efficacy of machine learning in XSS detection, limitations such as class imbalance in real-world scenarios and computational costs were noted. Future research will be focused on handling imbalanced data and practical deployment for better scalability and applicability in real-world scenarios. This work highlights the potential of machine learning in providing reliable and scalable solutions for network security.

**Keywords**: Cross site scripting attacks, Random Forest, Gradient Boosting, XGBoost, precision, recall, F1-score, etc.

# **Introduction**

Cross-Site Scripting or XSS represents a significant risk to the security of web applications, in this type of attacks an attacker injects malicious scripts into benign Web pages and thus causing unauthorized disclosure or breaching user trust (Tripathi and Thingla, 2019). These are among the most prevalent security challenges that one is mostly running into within the present scenario of ever-expanding digital dimensions. Therefore, focuses on building up a machine learning framework that will classify the network traffic as benign or indicative of XSS attacks is very important. This project uses ensemble and boosting for enhancing the detection accuracy by reducing false-positives and hence guaranteeing protection to web applications.

The selected dataset includes approximately 8.5 million labeled network traffic samples, out of which 6,099,469 samples are benign and 2,455,020 samples are XSS. The project is designed to use advanced machine learning techniques such as Random Forest, XGBoost and Gradient Boosting with the aim of building a reliable model for the detection of XSS attacks. Results will be evaluated against the set performance metrics, which are precision, recall, accuracy and F1 score for measuring the viability of the models. This comparative analysis against existing literature should contribute further to the advancement and state of the art for this field in web applications security.

## **Background**

Web applications form part and parcel of modern communication, commerce and even day-to-day activities, hence, they have been a prime target for cyber-attacks. Among many different vulnerabilities, XSS attacks stand out with a high level of simplicity and probably devastating results (S, 2021). Such attacks enable the injection of malicious scripts into user input by exploiting some weakness in input validation which will later get executed within a victim's browser. The consequences of a successful XSS attack include data theft, hijacking of sessions, defacement of web pages and malware propagation.

Traditional detection mechanisms either rule-based or signature-based can hardly cope with the dynamic nature and constantly gained sophistication of XSS attacks. In this regard, the machine learning approach has represented a very promising solution regarding how to address these new challenges by offering the capabilities needed to analyze complex patterns in network traffic and detect anomalies indicative of malicious activity (S, 2021).

This project leverages studies that have been previously conducted and which showed that feasibility of machine learning within the field of web application security. It also draws great background work done in many other works related to the efficiency of ensemble and boosting approaches against the detection of cyber-attacks, hence forming a concrete backbone for a proposed classification framework. Advanced techniques will be used throughout this project to strike a perfect balance between high rates of detection and low positives in false cases, which have become so critical in real-world scenarios for XSS attack detection.

Therefore, the focus of this project on the use of ensemble and boosting methods for the classification of XSS attacks is a big step in boosting the security of web applications. With a comprehensive dataset, robust evaluation metrics and a systematic approach, the proposed framework will be practical and effective in coping with this pervasive cybersecurity threat.

## **Related Work**

The studies reviewed from the paper “Machine Learning-Driven Detection of Cross-Site Scripting Attacks” (Rahmah Alhamyani and Majid Alshammari, 2024) indicate an increasing trend in the effectiveness of ML approaches to the detection of XSS attacks by virtue of their adaptability to evolving attack patterns and data-driven precision. Models such as Random Forest, SVMs and Gradient Boosting show high accuracy, with Random Forest achieving up to 98% in some implementations. Traditional boosting ensemble and hybrid methods achieve exceptional performance with XGBoost, in conjunction with feature selection techniques. Recent deep learning-based models involving CNNs, LSTMs and their hybrid forms have taken this further in the direction of perfecting the detection of malware with high accuracy of above 99% in some cases. Novel frameworks that have been proposed based on an attention mechanism, GANs and reinforcement learning increase robustness in detecting the presence of adversaries and against evolving attacks. Besides, applications vary from IoT to cloud-based environments showing the flexibility of the ML-based XSS detection methods that can assure realistic web security solutions. These bring us to the important role played by preprocessing, feature selection and hybrid approaches in optimizing ML models for a reliable XSS attack detection methodology.

The reviewed studies from the paper “Detection of cross-site scripting (XSS) attacks using machine learning techniques: a review” (Kaur, Garg and Bathla, 2023) have reported important advances in the detection of XSS attacks using machine learning techniques. Detection accuracy as high as 95% and above 99% has been achieved by researchers using classifier models such as SVMs, Random Forest, J48 and Gradient Boosting. Feature engineering is essential for enhancing detection performance. Approaches based on JavaScript events, URL attributes and HTML tags prove to be effective in the detection of drive-by download attacks. Some works have emphasized that t-SNE and Word2Vec are effective for the optimization of data representation. Ensemble methods such as Adaptive Boosting and Random Forest showed very promising accuracy with a low false-positive rate and proved suitable for deployment in real-time applications. It can be seen that combined hybrid approaches of machine learning, along with traditional web security testing tools or feature extraction techniques, show better success in the detection rate and also reducing computational overheads. Tree-based classifiers work particularly fast, such as Decision Trees balanced by their simplicity and speed of performance. Other relevant topics, such as attacks related to vulnerabilities in social networking applications or those involved in OAuth workflows, were also covered, exemplifying how ML fits into various roles related to web applications. These findings give the potential of advanced machine learning methodologies for achieving robust and scalable XSS attack detection systems.

The findings from the reviewed studies in the paper “Machine and Deep Learning-based XSS Detection Approaches: A Systematic Literature Review” (Thajeel et al., 2023) infuse the evolution of the methods for the detection of XSS attacks, which starts with traditional methods to machine learning as well as deep learning approaches. The reviews give an indication that there is reliance on both static and dynamic analysis methods, with client-side solutions being predominant and lacking adaptability to new patterns of XSS attacks. Various research has categorized existing approaches according to their place of implementation, including static analysis, dynamic analysis and hybrid methods, with trends shifting gradually toward the combination of traditional techniques with machine learning for higher accuracy. Other reviews pinpoint the strengths in supervised and unsupervised learning, though some have cited a lack of exploration towards semi-supervised, reinforcement and deep learning models. Surveys show that traditional approaches are still more widespread than AI-driven ones, although the most recent works prove the potential of hybrid methods that include static analysis, genetic algorithms and ML. Most noticeably, in these reviews one finds the absence of preprocessing, feature selection and evaluation stages within all the AI-based detection approaches of XSS attacks. Therefore, the need has arisen for a complete framework for enhancing robustness and efficiency in Machine Learning/Deep Learning to perform XSS detection under a wide variety of real web application contexts.

# **Methodology**

This project categorizes the network traffic using a systematic approach either as benign or indicative of XSS attacks by employing machine learning approaches. The methodology here will be prepared based on data preparation, model development and evaluation for strong analysis and practical applicability.

## **Dataset Used**

The dataset used for this project is sourced from [NF-UQ Dataset Repository](https://rdm.uq.edu.au/files/a4ad7080-ef9c-11ed-a964-b70596e96ad5). This dataset contains labeled network traffic samples categorized into two classes: benign and XSS.

* **Total Samples:** Approximately 8.5 million
  + **Benign Samples:** 6,099,469 (normal and unmalicious flows)
  + **XSS Samples:** 2,455,020 (malicious flows representing XSS attacks)
* **Dataset Characteristics:**
  + Contains features from network flow including packet size, duration and direction of flow.
  + Pre labeled for supervised learning.
  + The class imbalance is very high, 71.5% are benign and 28.5% are XSS. So balancing techniques must be used.

## **Data Processing**

* **Preprocessing:**
  + **Cleaning:** Removing entries not corresponding or missing.
  + **Feature Encoding:** Representing the categorical variables in numeric format.
  + **De-duplication:** Removal of duplicate samples.
  + **Balancing:** Drawing random 50,000 samples of each class to reduce computation complexity without loss of representation during model training.
* **Data Split:**
  + **Training Set:** 80% of the data for model training and tuning.
  + **Testing Set:** This would contain 20% of the data for performance evaluation.

## **Model Development**

Classification task is performed by leveraging the following machine learning techniques:

* **Ensemble Methods:**
  + **Random Forest:** An effective ensemble technique that generates multiple decision trees during the training phase and integrates their predictions to enhance the model's overall performance while minimizing the risk of overfitting (Banerjee et al., 2020). It works by splitting data at decision nodes based on features and averaging predictions of all trees for a final classification.
* **Boosting Methods:**
  + **Gradient Boosting:** It is an iterative boosting technique that builds weak learners in sequence, usually a decision tree, to improve mistakes of previous models (None Priya Karkare, 2024). While increasing the emphasis on misclassified samples with each iteration, Gradient Boosting improves predictive accuracy incrementally.
  + **XGBoost:** This is an advanced version of Gradient Boosting with improved speed and performance. Some of the feature inclusions are tree pruning, regularization and parallel processing, which makes this algorithm very efficient and great to work with on huge sets of data with high-dimensional feature space (Sharma and Yadav, 2023).

## **Evaluation Metrics**

The models' effectiveness is assessed through commonly used classification metrics:

* **Accuracy:** This quantifies the proportion of samples that are classified correctly.
* **Precision:** It shows the ratio between true positives and the sum of predicted positives. Higher precision would mean greater reliability of the detection of XSS.
* **Recall (Sensitivity):** Evaluates the proportion of correctly identified positives relative to the total number of actual positives, focusing on the model's effectiveness in identifying XSS attacks.
* **F1 Score:** The F1 score represents the harmonic mean of precision and recall, effectively striking a balance between minimizing false positives and false negatives.
* **Confusion Matrix:** This is an elaborated form of classification performance wherein the pattern of misclassifications could be presented.

## **Statistical and Data Mining Instruments**

* **Feature Selection:** Mutual information and other statistical measures will be used to identify the most important features for classification, hence reducing dimensionality and enhancing model efficiency.
* **Hyperparameter Tuning:** Techniques like grid search and randomized search will be used to optimize the model parameters for the best performance.
* **Data Mining Tools:** Preprocessing and analyses will be done using python libraries such as scikit-learn, numpy and pandas. XGBoost’s built-in library will be used for implementing boosting methods.

# **Results**

The results section encompasses all findings from the analytical effort to detect attacks, as per the network traffic records. Key metrics and pattern detection were made from this dataset in terms of various statistical summaries, features' importance, and performances by different models. Considering the below results, one could understand how a given model of machine learning performs a better job in distinguishing whether a certain network traffic is a positive or negative one based on the characteristics.

A screenshot of a computer code

Description automatically generated

Figure : Datatypes of variables

This dataset contains **100,000 rows** and **36 columns**, primarily numerical data with three columns of type float64 and the rest as int64. Key attributes include network traffic metrics (e.g., IN\_BYTES, OUT\_BYTES), TCP flags, flow durations, packet statistics, and throughput measures. The target variable is Attack, which likely indicates whether the data represents a benign or malicious activity. The data is clean with no missing values, making it ready for analysis.

A graph with lines and dots

Description automatically generated

Figure : Bytes and Packet length based on attack

The malicious traffic is less variable and with much smaller values for inbound bytes than the benign traffic does, with noticeable outliers when the count of bytes is higher. Similar to IN\_BYTES, outbound bytes are smaller and more consistent in malicious traffic, while benign traffic has larger outliers and higher variability. Minimum packet length is always higher in malicious traffic with less variability than benign, which has a greater range and lower median.

A graph of different colored bars

Description automatically generated

Figure : Correlation of variables

The plot shows the top 10 most strongly correlated features with the Attack variable, with MIN\_IP\_PKT\_LEN having the highest value (~0.7), implying it might contain critical information in distinguishing malicious traffic. Other features, like PROTOCOL, MAX\_IP\_PKT\_LEN, and LONGEST\_FLOW\_PKT, have strong positive values, indicating that packet size and protocol-related measures have an important role. Additionally, it will feature such as CLIENT\_TCP\_FLAGS, TCP\_WIN\_MAX\_IN, and TCP\_WIN\_MAX\_OUT are contributing to this, pointing out their relevance for modeling malicious activity. Besides this, other features such as MIN\_TTL, MAX\_TTL, and SRC\_TO\_DST\_AVG\_THROUGHPUT express information on the pattern of time-to-live and throughput, making them indispensable in any predictive model.

## Model Results

* **Random Forest**

**A screenshot of a graph

Description automatically generated**A blue squares with white text

Description automatically generated

Figure : Random Forest performance

The Random Forest Classifier yields an exceptional performance with 99.73% overall accuracy against the dataset. The resultant confusion matrix has shown only minimal misclassifications, comprising of 65 benign classified as XSS and only 16 XSS classified as benign, out of the total of 30,000 samples. The classification report also confirmed strong metrics for both classes with a value of 99.73% for precision, recall, and f1-score, hence showing that this model is able to successfully classify benign and malicious XSS traffic with high reliability and balanced performance among classes.

* **Gradient Boosting**

**A screenshot of a graph

Description automatically generated**A blue squares with white text

Description automatically generated

Figure : Gradient Boosting performance

The Gradient Boosting model works amazingly, with an accuracy of 99.51%. From the confusion matrix, there is a slight increase in misclassification compared to Random Forest: it misclassifies 94 benign as XSS and 53 XSS as benign out of the total 30,000 samples. Also, the classification report also shows high precision, recall, and f1-score for both classes, standing at 99.51% each. While slightly less accurate than Random Forest, this model has performed robustly-being able to distinguish between benign and XSS traffic.

* **XGBoost classifier**

**A screenshot of a report

Description automatically generated**A diagram of a confusion matrix

Description automatically generated

Figure : XGBoost performance

XGBoost is doing an excellent job with its overall accuracy of 99.75%, a little higher compared to the Random Forest and Gradient Boosting. Its confusion matrix is small in numbers of misclassifications-it has only 65 samples as XSS that are misclassified from benign, while a total of 10 from XSS are misclassified into benign out of 30,000 total samples. The classification report has almost perfect precision, recall, and f1-score, at 99.75% for both classes, showing that the classifier is very effective in segregating benign and malicious XSS traffic. Thus, XGBoost is the best model among these three.

## Tuned Performance

* **Random Forest**

**A screenshot of a report

Description automatically generated**A blue squares with white text

Description automatically generated

Figure : Random Forest tuned performance

The tuned Random Forest model achieves an impressive accuracy of **99.71%**, closely matching the original Random Forest model's performance. The confusion matrix shows minor misclassifications, with 69 benign samples misclassified as XSS and 18 XSS samples misclassified as benign out of 30,000 total samples. The classification report highlights excellent precision, recall, and f1-score for both classes at **99.71%**, reflecting the model's high capability in accurately distinguishing benign and malicious (XSS) traffic.

* **Gradient Boosting**

**A screenshot of a graph

Description automatically generated** A blue squares with white text

Description automatically generated

Figure : Gradient Boosting tuned performance

The tuned gradient boosting achieves a classification accuracy of 99.72%, so one could easily recognize from this that performance for tuned Gradient Boosting is quite high. In the confusion matrix for a tuned model, only 72 out of 30,000 total samples were misclassified as XSS, where only 12 out of these were XSS. Therefore, misclassifications in benign and XSS are quite negligible, which can be seen from this value. The classification report has specified high precision, recall, and f1-score at values of 99.72% for both classes, showing the robustness in the capability of distinguishing between benign and malicious/XSS traffic. Tuning improved this model to give more accurate predictions.

* **XGBoost classifier**

**A screenshot of a report

Description automatically generated**A diagram of a confused matrix

Description automatically generated

Figure : XGBoost tuned performance

The tuned model of XGBoost shows a very impressive accuracy of 99.74%, with slight improvements from the untuned model. The confusion matrix presents a minimum misclassification: 68 benign samples misclassified as XSS, and 10 XSS samples misclassified as benign out of 30,000 total samples. The classification report also looks very good, where precision, recall, and f1-score for both classes are 99.74% each. The tuning process increases the performance of the model by a high margin, which proves to be very effective in differentiating benign and malicious (XSS) traffic.

## Cross validation performance

|  |  |
| --- | --- |
| Default model | |
| Random Forest (Default) Cross-Validation Accuracy | 0.9966 ± 0.0006 |
| Gradient Boosting (Default) Cross-Validation Accuracy | 0.9948 ± 0.0008 |
| XGBoost (Default) Cross-Validation Accuracy | 0.9969 ± 0.0005 |
| Tuned model | |
| Random Forest (Tuned) Cross-Validation Accuracy | 0.9967 ± 0.0007 |
| Gradient Boosting (Tuned) Cross-Validation Accuracy | 0.9966 ± 0.0007 |
| XGBoost (Tuned) Cross-Validation Accuracy | 0.9969 ± 0.0005 |

Table : Cross validation performance

Cross-validation results indicate that all three models, namely Random Forest, Gradient Boosting, and XGBoost, have high accuracy with minimal variance, which is indicative of their performance. Among the default models, the best performance is given by XGBoost, with an accuracy of 99.69% ± 0.0005, closely followed by Random Forest (99.66% ± 0.0006) and Gradient Boosting (99.48% ± 0.0008). After tuning, XGBoost stays the best with 99.69% ± 0.0005, while both RandomForest and Gradient Boosting improved a bit: 99.67% ± 0.0007 and 99.66% ± 0.0007 correspondingly. This means XGBoost is the most robust model, even without extensive tuning.

# **Conclusion**

The analysis done proved that all three machine learning models, namely, Random Forest, Gradient Boosting, and XGBoost, fared exceptionally well in identifying malicious traffic. The most robust model was XGBoost, with an accuracy of 99.74% after tuning. Even Random Forest and Gradient Boosting fared well, proving their worth in segregating benign from malicious-XSS traffic.

These features are very important, including MIN\_IP\_PKT\_LEN and PROTOCOL, which show a very good correlation with the target variable, hence useful for predictions by the models. Model tuning slightly improved the performance, most especially for Gradient Boosting, which closed up on the other two models. Cross-validation results confirmed that the models were also consistent and reliable across different subsets of the data.

However, there are a number of limitations to this study. First, all the proposed models are evaluated based on balanced data; hence, it may not reflect actual traffic congestion scenarios with imbalance in classes. Second, the computational cost could be very high to tune some complex models such as XGBoost.

In the future, one could apply techniques like undersampling, oversampling, or cost-sensitive learning to deal with imbalanced datasets. Besides that, real-world deployment of the best performing model may be able to show real-world utility and efficiency. Generally, this study has shown how machine learning can be used effectively and reliably for malicious network traffic detection.

# **References**

Banerjee, R., Baksi, A., Singh, N. and Bishnu, S.K. (2020). *Detection of XSS in web applications using Machine Learning Classifiers*. [online] IEEE Xplore. doi:https://doi.org/10.1109/IEMENTech51367.2020.9270052.

Kaur, J., Garg, U. and Bathla, G. (2023). Detection of cross-site scripting (XSS) attacks using machine learning techniques: a review. *Artificial Intelligence Review*. doi:https://doi.org/10.1007/s10462-023-10433-3.

None Priya Karkare (2024). The Artistry of Stochastic Gradient Boosting and Gradient Boosting Classifiers in Chronic Renal Disease Classification. *Deleted Journal*, 20(3), pp.2405–2415. doi:https://doi.org/10.52783/jes.4086.

Rahmah Alhamyani and Majid Alshammari (2024). Machine Learning-Driven Detection of Cross-Site Scripting Attacks. *Information*, [online] 15(7), pp.420–420. doi:https://doi.org/10.3390/info15070420.

S, H. (2021). Cross Site Scripting Attack Detection in Web using Machine Learning Algorithm. *International Journal for Research in Applied Science and Engineering Technology*, 9(4), pp.1542–1547. doi:https://doi.org/10.22214/ijraset.2021.33921.

Sharma, S. and Yadav, N.S. (2023). A Grey Wolf Optimized XGboost-Multilayer Stacking Approach to Detect XSS Attacks. *IETE Journal of Research*, 70(6), pp.5633–5648. doi:https://doi.org/10.1080/03772063.2023.2264251.

Thajeel, I.K., Samsudin, K., Hashim, S.J. and Hashim, F. (2023). Machine and Deep Learning-based XSS Detection Approaches: A Systematic Literature Review. *Journal of King Saud University - Computer and Information Sciences*, [online] 35(7), p.101628. doi:https://doi.org/10.1016/j.jksuci.2023.101628.

Tripathi, P. and Thingla, R. (2019). Cross Site Scripting (XSS) and SQL-Injection Attack Detection in Web Application. *SSRN Electronic Journal*. doi:https://doi.org/10.2139/ssrn.3356292.