Contents lists available at IdNNS
Indonesian Journal of Cybersecurity and Emerging Risks
Volume 1 ; Issue 1
journal homepage: https://journal.idnns.org/index.php/ijcyber

# Machine Learning-Based Detection of Cross-Site Scripting (XSS) Attacks Using Classification Algorithms

Anupama Mishra[b], Pramod Kumar[a,*]

[b]Computer Science & Engineering, School of Science & Technology, Swami Rama Himalayan University, Dehradun, Uttarakhand, 248016, India
[a]School of Science & Technology, Swami Rama Himalayan University, Dehradun, Uttarakhand, 248016, India

ARTICLE INFO

*Keywords*:
Cross site scripting
Cyber Security
Machine learning
Web Security

ABSTRACT

Cross-Site Scripting (XSS) attack is the most dominant attack in web applications. Its empowering attackers to inject malicious scripts into trusted web environments. The research paper proposed a machine learning-based framework for the detection and classification of XSS attacks. We use the dataset from Kaggle that contains textual web request patterns labeled as normal and malicious instances. Experimental analysis was performed to evaluate the effectiveness of multiple supervised learning algorithms including Logistic Regression, Random Forest, and Support Vector Machine. The methodology involved preprocessing, feature extraction, model training, testing, and performance evaluation parameters for binary classification.

## 1. Introduction

We live in the digital era of modern digital ecosystems that support e-commerce, online banking, healthcare systems, educational platforms, and cloud-based services through web applications. However, the development of web technologies, especially due to the inclusion of artificial intelligence tools Younas et al. (2024), has exponentially increased. This convenience and luxury do not come alone; these platforms are also exposed to significant cybersecurity threats, particularly Cross-Site Scripting (XSS) attacks.

XSS is a type of injection attack that involves injecting malicious scripts into benign and trusted websites. It targets vulnerable websites in the form of browser-side scripts to execute malicious code on different end-users' browsers. A website's vulnerability allows a web application to use inputs from a user within the generated output without proper validation or encoding OWASP Foundation (2024).

A malicious script is sent to the trusted users by performing an XSS attack. XSS sits between the browser and the web applications; from there, it injects the suspicious code and either retrieves/steals sensitive information for the attacker's benefit or damages resources. Due to trust, the end user's browser executes the script. The malicious script can access any cookies, session tokens, or other private data stored by the browser and used with that website, since it believes the script originated from a reliable source. Even the HTML page's content may be altered by these scripts.

According to the Open Web Application Security Project (OWASP) OWASP Foundation (2024), XSS continues to remain among the most critical web application vulnerabilities due to its widespread exploitation and severe security implications. Rule-based filters and signature-based Web Application Firewalls (WAFs) sometimes fail to detect polymorphic or obfuscated XSS payloads. Therefore, researchers have continuously worked on and explored technologies like machine learning and deep learning to address this issue.

Intelligent approaches can learn the patterns in malicious scripts and classify unknown attack payloads more effectively than static rule-based systems. Recent research studies have substantially improved the detection of XSS using supervised classification learning models Hu et al. (2023); Li et al. (2025).

In this research paper, we focus on multiple machine learning algorithms for the detection of XSS attacks. The major contributions of this study are as follows:

1. Development of a machine learning-based framework for XSS attack detection.
2. Comparative analysis of multiple classification algorithms using Data Mining.
3. Evaluation of classification performance using standard performance metrics.
4. Identification of the most effective model for detecting malicious web payloads.

The rest of this paper is structured as follows: Section 2 presents the literature review, Section 3 presents the proposed

*Corresponding author
anupamamishra@srhu.edu.in (A. Mishra); tulasacademic@gmail.com (P. Kumar)
ORCID(s):

**Table 1**
Comparative Analysis of Literature Review

| Authors & Ref. | Technique / Methodology / Architecture | Feature Engineering | Outcomes | Limitations |
|---|---|---|---|---|
| Younas et al. Younas et al. (2024) | Framework based on ML and DL using Random Forest and LSTM | TF-IDF feature extraction from XSS payloads | Random Forest achieved nearly 99% detection accuracy, outperforming other models | High computational dependency for deep learning models; performance may vary with dataset diversity |
| Hu et al. Hu et al. (2023) | Two-channel feature fusion framework using CNN, Bi-LSTM, and Self-Attention | Semantic and contextual payload representation | Achieved high precision and recall in XSS classification | Increased architectural complexity and training cost |
| Li et al. Li et al. (2025) | Hybrid Deep Learning Architecture combining CNN, sequential learning, and attention mechanisms | Deep semantic feature extraction from payload sequences | Significant improvement in attack classification performance | Requires large training datasets and high computational resources |
| Santithanmanan et al. Santithanmanan et al. (2023) | Machine Learning classifiers including Gaussian Naïve Bayes and $k$-NN | URL syntactic feature extraction | ML approaches effectively classified malicious URLs | Limited capability for detecting highly obfuscated payloads |
| Miczek et al. Miczek et al. (2025) | Large Language Models (LLMs) for adversarial payload generation | Obfuscated and adversarial XSS payload generation | Demonstrated enhanced model robustness against obfuscated attacks | LLM-generated samples may introduce training bias and increased preprocessing complexity |
| Ayoubi et al. Ayoubi et al. (2026) | Hybrid CNN-SVM framework | Grayscale image representation of payloads | Reported good performance with lower resource consumption | Image conversion may lose semantic textual information |

methodology, Section 4 describes the experimental setup, Section 5 discusses results, and Section 6 concludes our work.

## 2. Literature Review

Many researchers have studied XSS attacks by applying machine learning and deep learning techniques to enhance web application security.

Younas et al. proposed a machine learning and deep learning framework for early XSS detection using TF-IDF feature engineering and Long Short-Term Memory (LSTM) models. The authors reported a detection accuracy of nearly 99% Younas et al. (2024).

Hu et al. introduced a two-channel feature fusion framework combining Convolutional Neural Networks (CNN), Bidirectional Long Short-Term Memory (Bi-LSTM), and self-attention mechanisms for XSS detection Hu et al. (2023). The proposed architecture effectively captured both local and contextual semantic features from malicious payloads and achieved high precision and recall.

Li et al. Li et al. (2025) proposed a hybrid deep learning architecture for identifying complex XSS attack patterns. Their research showed that combining convolutional neural layers with sequential learning and attention mechanisms can significantly improve attack classification performance.

Santithanmanan et al. discussed in Santithanmanan et al. (2023) the application of machine learning algorithms such as Gaussian Naïve Bayes and $k$-Nearest Neighbor ($k$-NN) classifiers for URL-based XSS attack detection. Their outcomes indicated that ML-based approaches can effectively classify malicious URLs based on extracted syntactic features.

Existing and recent research has also discussed adversarial and obfuscated XSS payload detection. Miczek et al. presented the use of Large Language Models (LLMs) for generating obfuscated XSS samples to improve machine learning robustness Miczek et al. (2025). Ayoubi et al. proposed a hybrid CNN-SVM architecture using grayscale image representations of payloads to identify XSS attacks Ayoubi et al. (2026). Their research reported good performance while keeping the computational overhead low.

Although there are many existing mechanisms to detect XSS attacks, they have limitations in terms of computational resources, dataset size, ML algorithm selection, and preprocessing methods. To address these issues, we propose

a machine learning-based framework for XSS detection. Additionally, we present a comparative analysis using visual data.

Table 1 presents a summary of the comparative analysis based on the literature review. The table shows the development of technologies and methodologies to detect XSS attacks and outlines key limitations that need to be addressed.

## 3. Methodology

The proposed work includes preprocessing, selection of machine learning algorithms Rathore and Sharma (2017); Thajeel et al. (2023); Babaey and Ravindran (2025), and evaluation of the algorithm after ML modelling. The methodology has the following key factors:

### 3.1. Data Description

We used XSS dataset that consist of web payload samples labelled into malicious and benign classes. The dataset contains attributes Hussain (2023) as Sentence: Web payload or script content and Label as 0 for Benign and 1 for Malicious XSS. The size of the dataset is 13,686. To reduce high dimensional TF-IDF vectors, we applied Principal Component Analysis. It reduces the vectors into two dimensions. It shows that TF-IDF character n-grams captured discriminative attack patterns effectively. In figure 1 the data set is projected on to two principal components pca component 1 and pca component 2, enabling visual interpretation of the data distribution and class separation. The visualization shows two distinct clusters represented by two different colours indicating the classification of normal and malicious instances. The purple colour cluster is concentrated on the left side of the graph where is the yellow coloured cluster distributed right side of the graph. This clear separation shows the extracted features possess strong discriminatory characteristics, allowing machine learning algorithms to effectively differentiate between different classes. Additionally, the limited overlap between the clusters indicates reduced ambiguity and improved classification capability Mishra et al. (2024); Kshetri et al. (2024). Therefore, the PC visualization confirms that the data set is well structured and suitable for classification by applying machine learning algorithms.

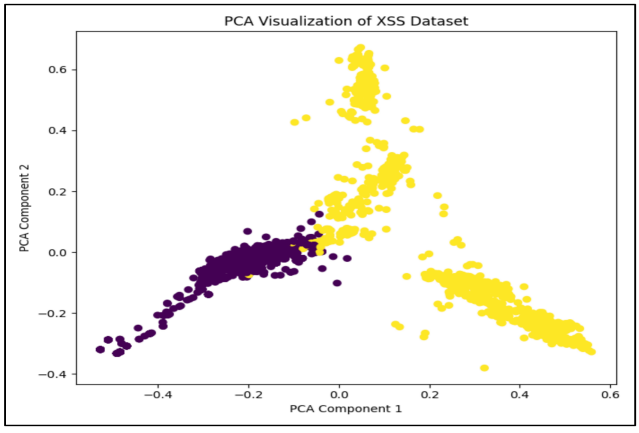


Figure 1: PCA visualization of dataset

Now, we check the issue of imbalance of the dataset. The distribution of the binary classes can be seen in figure 1. It can be interpreted that both the classes are having a good no of samples so there is no issue of imbalance dataset. This dataset is sufficiently enough for the experiment. Therefore, we can now apply the ML algorithm for building the model.

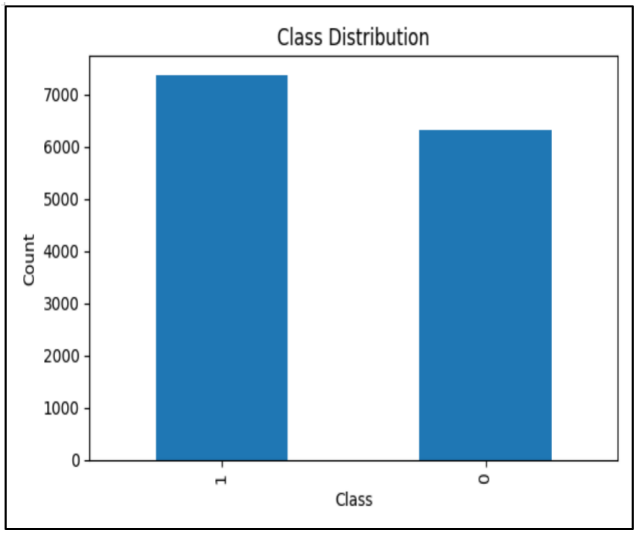


Figure 2: Class Distribution

### 3.2. Data Preprocessing

Since XSS payloads often contain irregular scripts encoded characters, symbols and malicious patterns Mishra et al. (2021); Mishra and Colace (2022). Multiple preprocessing operations were performed before feature extraction and classification. The operations are as following:

1. **Lowercase Transformation:** In this operation, all textual payload converted into lowercase to maintain textual consistency and reduce duplicate token representations to ensure uniformity and consistency across the data set. This step eliminates duplicate representation of similar tokens caused by case sensitivity for example terms such as `<SCRIPT>`, `<script>`, and `<Script>` are treated as the same token after transformation.
2. **Stopword Filtering:** Common English stopwords were removed to reduce irrelevant textual noise and improve discriminative feature learning. Words like “the”, “is”, “and”, “to”, “etc.” were removed from the data set because of minimal semantic contribution for a discriminatory value to detect the attack. Eliminating these frequently occurring but non informative words reduces irrelevant textual noise decreases feature space dimensionality and enables the model to focus on more meaningful attack related patterns.
3. **Tokenization:** Tokenized into smaller textual units for feature extraction and help in identification of attack signatures. It breaks complex payloads into interpretable components such as keywords symbols tags and script fragments. This process facilitates feature extraction and helps identify suspicious script structures commonly presents in XSS attacks including `alert()`, `<script>`, On error if rame and javascript event handlers.

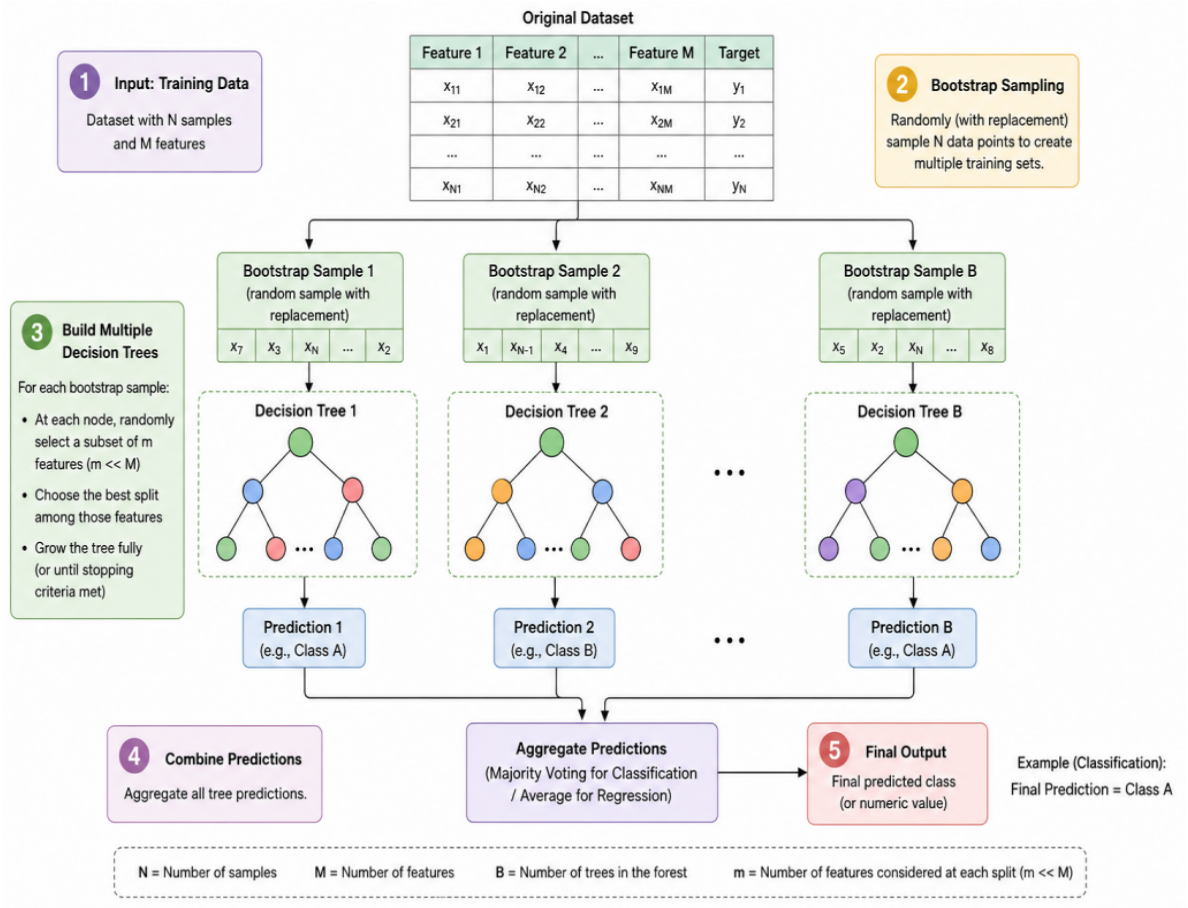


**Figure 3:** Working of Random Forest

4. **Regular Expression (Regex) Based Filtering:** Regex-based filtering was applied to preserve important special characters and scripting patterns commonly associated with XSS attacks. This helps to preserve critical symbols, scripting patterns, and malicious character sequences frequently associated with XSS attacks. Unlike traditional text cleaning methods that remove special characters. This approach selectively retains meaningful attack specific patterns such as:
   - HTML Tag(`<script>`, `<iframe>`)
   - Javascript functions(`alert()`, `eval()`)
   - Encoded payloads
   - Special Symbols(`<`, `>`, `(`, `)`, `,`, `'`, `"`)

   Preserving such patterns is essential because these symbols often serve as indicators of malicious payload execution behaviour.
5. **Document Frequency Filtering:** Extremely frequent tokens were filtered using relative document frequency thresholds to reduce dimensionality and improve model generalization. Tokens With excessively high occurrence rates generally provide limited risk discriminative capability because they appear in both benign and malicious samples. Removing these overly common tokens helps reduce dimensionality, minimises over fitting, and enhances the generalisation capability of the classification models.
6. **TF-IDF Feature Extraction:** After preprocessing, the textual payloads were transformed into numerical vectors using TF-IDF representation. The TF-IDF scheme assigns higher importance to terms that are:
   - frequent within a payload
   - rare across the dataset

   This improves the detection of malicious XSS patterns. Terms that occur frequently in a specific payload but in frequently in a specific payload across other payloads receive higher weights, Making them highly informative for classification. In the context of XSS detection, Malicious scripting keywords an attack patterns obtain greater significant compared to common benign terms. Consequently, TF-IDF improves feature discrimination, enhances attack pattern recognition, and increases the overall detection performance of machine learning models.

### 3.3. Machine Learning Algorithms

We have applied theses machine learning algorithms as Random Forest, Support Vector Machine, and Logistics regression. All the three are supervised classifiers and work

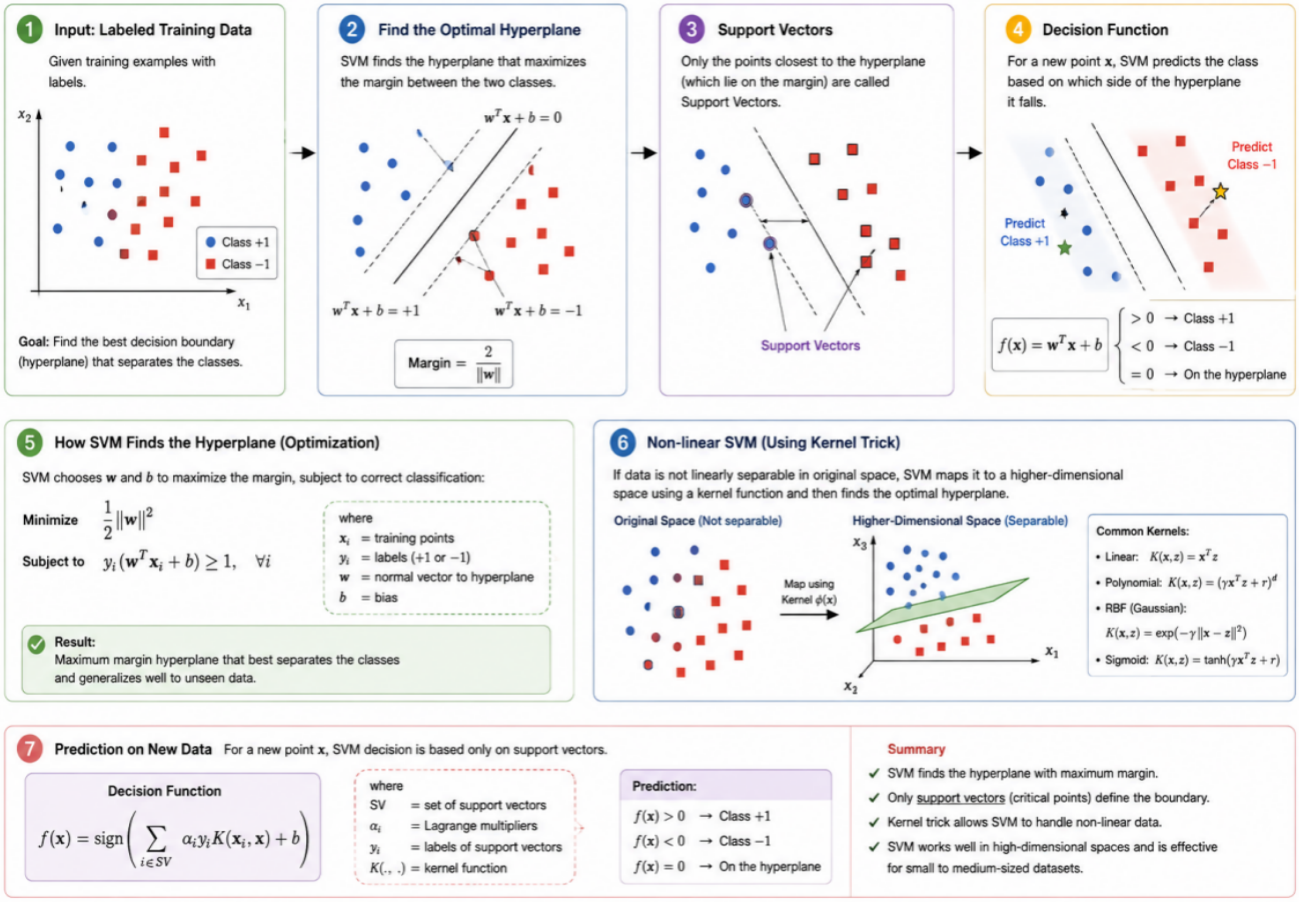


**Figure 4:** Working of Support Vector Machine

for both binary and metaclassifiers. In the construction of random forest, each tree in the forest is considered as random subset of the features. While Support Vector Machine draws a hyper plane and divide the datapoints into binary or multi classifications. While Logistics regression applies a sigmoid function and categorized the data points Cheng et al. (2025); Gharai et al. (2025); Wali et al. (2025); Shaisob et al. (2025).

1. **Random Forest:** Random forest is an ensemble supervised machine learning algorithm that constructs multiple decision trees using different random subsets of the training data set and features. Each tree independently predicts the output class and the final prediction is determined through majority voting in classification tasks. Figure 3 presents bootstrap sampling generates diverse datasets for individual trees which improves generalisation and reduce overfitting. The aggregation of prediction from multiple trees enhances classification accuracy robustness and stability making random forest highly suitable for cyber security data sets such as XSS attack detection where feature variability is high.
2. **Support Vector Machine:** Support vector machine is a supervised machine learning algorithm that identifies the optimal hyper plane separating different classes with the maximum margin. Figure 4 presents how support vectors which closest to data points to the decision boundary play a critical role in defining the classifier. SVM transforms a non linear data Into higher dimensional space based on kernel functions such as radial basis function enabling efficient classification of intricate attack patterns. Due to its strong capability in handling high dimensional Sparse datasets, SVM is widely adopted for intrusion direction malicious url direction and web attack classification.
3. **Logistics Regression:** Logistic regression is a probabilistic supervised classification algorithm used for predicting binary outcomes. As shown in figure 5 the algorithm first computes a linear combination of input features and then applies the sigmoid activation function to convert the output into a probability value between zero and one. A threshold value is used to assign the final class label the model parameters optimised using cross entropy loss and gradient descent. Logistic regression is computationally efficient interpretable and effective for identifying whether a request belongs to normal or malicious category in web security applications such as cross site attack.

## 4. Experimental Setup

The experimental analysis was conducted using python 3.10 .4 64 bit in a Jupiter notebook environment integrated with visual studio code. The system configuration consist of an 11th generation intel core I 5 1135G7 processor operating

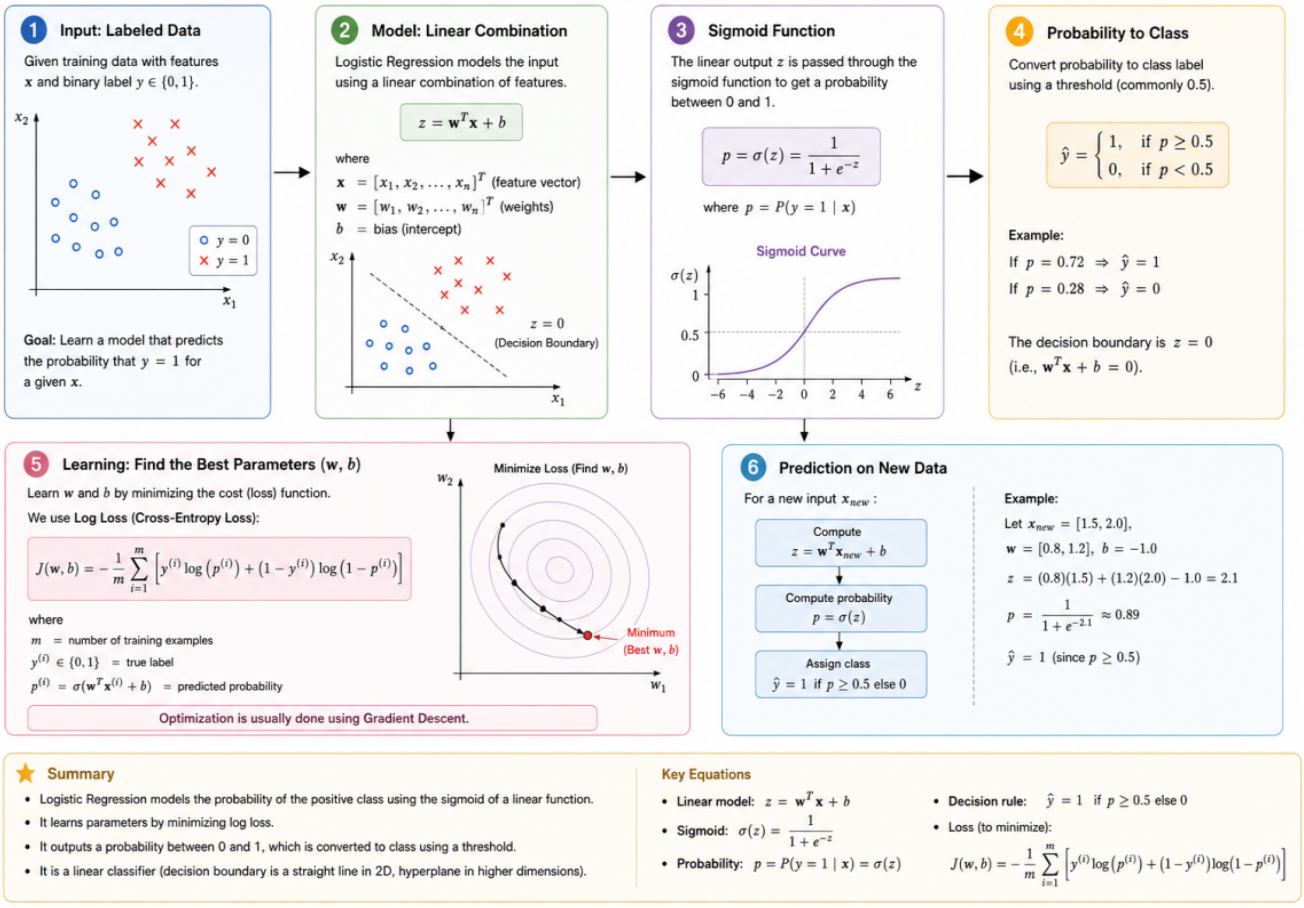


**Figure 5:** Working of logistic regression

at a clock speed of 2.40 gigahertz to 3.32 gigahertz Supported by 16 GBDDR 4RAM. The storage configuration included a 1TBHT along with a 256GB SSD To ensure efficient data processing and execution performance.

For model development training and Evaluation, several python based machine learning and data analysis were utilized primary including the sketch learnt framework the experimental workflow involved data set preprocessing, TF-IDF feature extraction, Dimensionality reduction, model training and performance evaluation Using standard classification Metrics such as accuracy, preseason, recall F1 score, ROC curve and AUC. The complete experimental setup ensure a stable computation Environmental for implementing and evaluating the random forest support vector machine and logistic regression model for excess attack detection.

## 5. Performance Evaluation Metrics

To evaluate the proposed machine learning models, several standard performance evaluation metrics were employed Bakır (2025); Meze et al. (2025); Zeng (2025).

Accuracy: The overall correctness of the classification model by calculating the ratio of correctly predicted instances to the total number of instances.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{1}$$

Precision: It evaluates the capability of classifier to correctly identify positive instances while reducing false positive predictions.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

High precision indicates that the model produces fewer false alarms and accurately identifies malicious payloads

Recall: Recall measures the ability of the classifiers to correctly detect actual positive instances.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

A high recall value signifies that the model successfully detects most malicious excesses attacks that with minimal false negatives.

F1 score: The F1 score provides a balanced measure between presion and recall by completing their harmonic mean.

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

This metric is particularly useful when dealing with imbalance data set.

**Table 2**
Evaluation Metrices of ML algorithms

| Model | Accuracy | F1 | Precision | Recall |
|---|---|---|---|---|
| Random Forest | 0.997 | 0.997 | 0.997 | 0.997 |
| SVM | 0.887 | 0.885 | 0.906 | 0.887 |
| Logistic Regression | 0.997 | 0.997 | 0.997 | 0.997 |

## 6. Result and Discussion

The results show and verify the applied methodology. In the experiment, we have applied three ML algorithms as Random Forest (RF), SVM and Logistics Regression (LR). Figure 6, figure 7 and figure 8 are refer to confusion matrix of RF, SVM and LR respectively. These figures show highly corrected numbers. Similarly, table 2 presents the performance metrices accuracy as 99.7%, 88.7%, and 99.7% for RF, SVM and LR respectively. It can be seen that RF and LR have performed well over SVM. The same results are visualized in figure 9. Receiver Operating Characteristics Riesthuis et al. (2025); Sant (2025); Deng et al. (2025) (ROC) are presented by figure 10 and 11 for the classes 0 and 1 respectively. It shows the ratio between TPR (True positive rate) and FPR (False positive rate). If the curve reaches top left corner, then it means that the performance of the model is good. Figure 10 presents the roc curve Comparison of Random Forest SVM and logistic regression classifiers. The curves for all three models are positioned near the upper left corner indicating high classification performance with low false positive rates in high true positive rates. Among the evaluated classifiers random forest in svm exhibits slightly superior performance Achieving better discrimination capability compared to logistic regression. The larger area under the roc curve presents that the Proposed models are highly effective for detection of XSS attack and its classification. Figure 11 presents all three models that indicating high true positive rate and low false positive rate. The Diagonal line performance of random classifier While the significant deviation of proposed model that confirms their strong predictive capability.

Among all classifiers, Random Forest produced the best overall performance in terms of accuracy and stability.

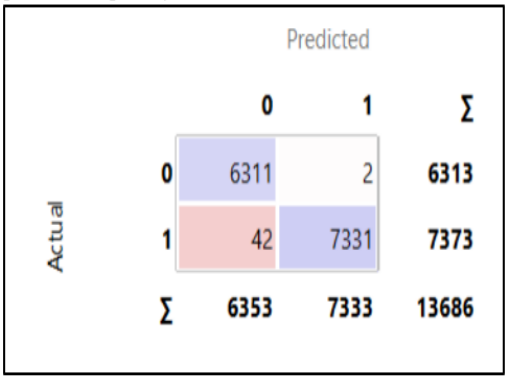


Figure 6: Confusion matrix for Random forest

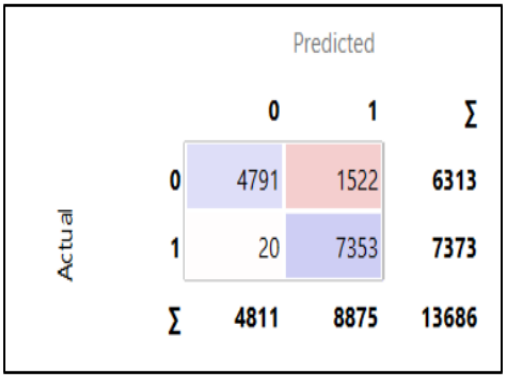


Figure 7: Confusion matrix for SVM

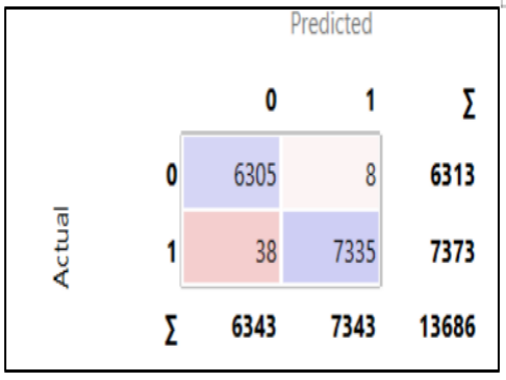


Figure 8: Confusion matrix for logistic regression

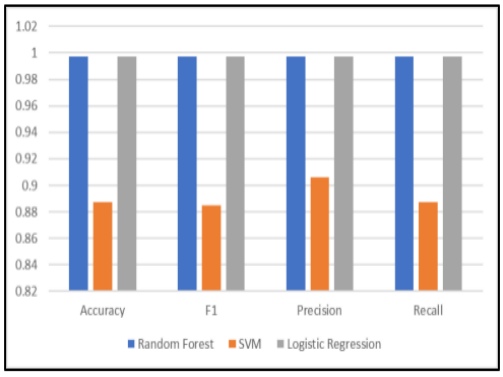


Figure 9: visualization of evaluation metrices of ML algorithms

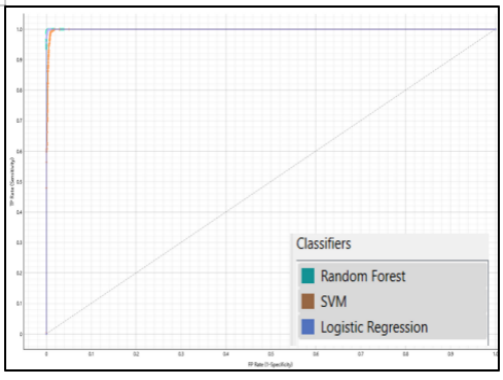


Figure 10: ROC for class 0

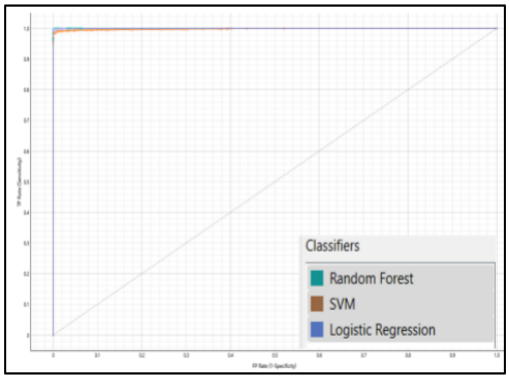


Figure 11: roc for class 1

## 7. Conclusion

In this research we have applied three algorithms as random forest, support vector machine and logistic regression. The dataset is taken from Kaggle platform. The results show that our work effectively able to classify malicious and benign payloads with high accuracy. Based on the results and ROC, it can be said that RF and LR both have performed well in comparison to SVM.

Due to AI involvement, the attackers are getting advanced and using AI based automated tools to launch the attack. Therefore, the future work involves deep learning to identify more complex and hidden attack patterns.

## References

Ayoubi, A., Laaouina, L., Jeghal, A., Tairi, H., 2026. An improved detection of cross-site scripting (xss) attacks using a hybrid approach combining convolutional neural networks and support vector machine. Journal of Cybersecurity and Privacy 6, 18.

Babaey, V., Ravindran, A., 2025. Genxss: An ai-driven framework for automated detection of xss attacks in wafs, in: SoutheastCon 2025, IEEE. pp. 1519–1524.

Bakır, R., 2025. Uniembed: A novel approach to detect xss and sql injection attacks leveraging multiple feature fusion with machine learning techniques. Arabian Journal for Science and Engineering 50, 15591–15604.

Cheng, S., et al., 2025. Machine learning for modelling unstructured grid data in computational physics: a review. Information Fusion 123, 103255.

Deng, J., et al., 2025. So you've got a high auc, now what? an overview of important considerations when bringing machine-learning models from computer to bedside. Medical Decision Making 45, 640–653.

Gharai, C., et al., 2025. Enhancing web security through machine learning-based feature selection for cross-site scripting (xss) attacks classification, in: 2025 IEEE 6th India Council International Subsections Conference (INDISCON), IEEE.

Hu, T., Xu, C., Zhang, S., Tao, S., Li, L., 2023. Cross-site scripting detection with two-channel feature fusion embedded in self-attention mechanism. Computers & Security 124, 102990.

Hussain, S.S., 2023. Cross-site scripting (xss) dataset. https://www.kaggle.com/datasets/syedsaqlainhussain/cross-site-scripting-xss-dataset. Accessed: 2026-06-02.

Kshetri, N., Kumar, D., Hutson, J., Kaur, N., Osama, O.F., 2024. algoxssf: Detection and analysis of cross-site request forgery (xsrf) and cross-site scripting (xss) attacks via machine learning algorithms. https://arxiv.org/abs/2402.01012. ArXiv Preprint arXiv:2402.01012.

Li, Z., Liu, F., Gu, Z., Liu, Y., 2025. Xss attack detection method based on cnn-bilstm-attention. Applied Sciences 15, 8924.

Meze, M., Bolojan, O.M., Costea, F.M., 2025. Comparison of machine learning algorithms used for detecting xss attacks, in: 2025 18th International Conference on Engineering of Modern Electric Systems (EMES), IEEE.

Miczek, D., Gabbireddy, D., Saha, S., 2025. Leveraging llm to strengthen ml-based cross-site scripting detection, in: Proceedings of the 2025 ACM Workshop on Wireless Security and Machine Learning, pp. 14–19.

Mishra, A., Colace, F., 2022. Cyber security traits of a future-ready organization. Cyber Security Insights Magazine, Insights2Techinfo 1, 16–19.

Mishra, A., Gupta, B.B., Santaniello, D., Mishra, K., 2024. Secure web applications based on moving target defense: Challenges, solutions, and new trends, in: Digital forensics and cyber crime investigation. CRC Press, pp. 1–16.

Mishra, A., Hsu, C.H., Arya, V., Chaurasia, P., Li, P., 2021. A hybrid approach for protection against rumours in a iot enabled smart city environment, in: International conference on cyber security, privacy and networking, Springer International Publishing, Cham. pp. 101–109.

OWASP Foundation, 2024. Cross site scripting (xss). https://owasp.org/www-community/attacks/xss/. Accessed: 2026-06-02.

Rathore, S., Sharma, P.K., 2017. Xssclassifier: An efficient xss attack detection approach based on machine learning classifier on snss. Journal of Information Processing Systems 13.

Riesthuis, P., Otgaar, H., Bücken, C., 2025. Ready to roc? a tutorial on simulation-based power analyses for null hypothesis significance, minimum-effect, and equivalence testing for roc curve analyses. Behavior Research Methods 57, 120.

Sant, L., 2025. Roc curves, in: International Encyclopedia of Statistical Science. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 2212–2215.

Santithanmanan, K., Kirimasthong, K., Boongoen, T., 2023. Machine learning based xss attacks detection method, in: UK Workshop on Computational Intelligence, Springer Nature Switzerland, Cham. pp. 418–429.

Shaisob, M.H., et al., 2025. Xss-safenet: A bidirectional lstm architecture for high-precision cross-site scripting detection, in: 2025 28th International Conference on Computer and Information Technology (ICCIT), IEEE.

Thajeel, I.K., Samsudin, K., Hashim, S.J., Hashim, F., 2023. Machine and deep learning-based xss detection approaches: a systematic literature review. Journal of King Saud University - Computer and Information Sciences 35, 101628.

Wali, S., Farrukh, Y.A., Khan, I., 2025. Explainable ai and random forest based reliable intrusion detection system. Computers & Security 157, 104542.

Younas, F., Raza, A., Thalji, N., Abualigah, L., Zitar, R.A., Jia, H., 2024. An efficient artificial intelligence approach for early detection of cross-site scripting attacks. Decision Analytics Journal 11, 100466.

Zeng, G., 2025. Invariance properties and evaluation metrics derived from the confusion matrix in multiclass classification. Mathematics 13, 2609.

## References

Ayoubi, A., Laaouina, L., Jeghal, A., Tairi, H., 2026. An improved detection of cross-site scripting (xss) attacks using a hybrid approach combining convolutional neural networks and support vector machine. Journal of Cybersecurity and Privacy 6, 18.

Babaey, V., Ravindran, A., 2025. Genxss: An ai-driven framework for automated detection of xss attacks in wafs, in: SoutheastCon 2025, IEEE. pp. 1519–1524.

Bakır, R., 2025. Uniembed: A novel approach to detect xss and sql injection attacks leveraging multiple feature fusion with machine learning techniques. Arabian Journal for Science and Engineering 50, 15591–15604.

Cheng, S., et al., 2025. Machine learning for modelling unstructured grid data in computational physics: a review. Information Fusion 123, 103255.

Deng, J., et al., 2025. So you've got a high auc, now what? an overview of important considerations when bringing machine-learning models from computer to bedside. Medical Decision Making 45, 640–653.

Gharai, C., et al., 2025. Enhancing web security through machine learning-based feature selection for cross-site scripting (xss) attacks classification, in: 2025 IEEE 6th India Council International Subsections Conference (INDISCON), IEEE.

Hu, T., Xu, C., Zhang, S., Tao, S., Li, L., 2023. Cross-site scripting detection with two-channel feature fusion embedded in self-attention mechanism. Computers & Security 124, 102990.

Hussain, S.S., 2023. Cross-site scripting (xss) dataset. https://www.kaggle.com/datasets/syedsaqlainhussain/cross-site-scripting-xss-dataset. Accessed: 2026-06-02.

Kshetri, N., Kumar, D., Hutson, J., Kaur, N., Osama, O.F., 2024. algoxssf: Detection and analysis of cross-site request forgery (xsrf) and cross-site scripting (xss) attacks via machine learning algorithms. https://arxiv.org/abs/2402.01012. ArXiv Preprint arXiv:2402.01012.

Li, Z., Liu, F., Gu, Z., Liu, Y., 2025. Xss attack detection method based on cnn-bilstm-attention. Applied Sciences 15, 8924.

Meze, M., Bolojan, O.M., Costea, F.M., 2025. Comparison of machine learning algorithms used for detecting xss attacks, in: 2025 18th International Conference on Engineering of Modern Electric Systems (EMES), IEEE.

Miczek, D., Gabbireddy, D., Saha, S., 2025. Leveraging llm to strengthen ml-based cross-site scripting detection, in: Proceedings of the 2025 ACM Workshop on Wireless Security and Machine Learning, pp. 14–19.

Mishra, A., Colace, F., 2022. Cyber security traits of a future-ready organization. Cyber Security Insights Magazine, Insights2Techinfo 1, 16–19.

Mishra, A., Gupta, B.B., Santaniello, D., Mishra, K., 2024. Secure web applications based on moving target defense: Challenges, solutions, and new trends, in: Digital forensics and cyber crime investigation. CRC Press, pp. 1–16.

Mishra, A., Hsu, C.H., Arya, V., Chaurasia, P., Li, P., 2021. A hybrid approach for protection against rumours in a iot enabled smart city environment, in: International conference on cyber security, privacy and networking, Springer International Publishing, Cham. pp. 101–109.

OWASP Foundation, 2024. Cross site scripting (xss). https://owasp.org/www-community/attacks/xss/. Accessed: 2026-06-02.

Rathore, S., Sharma, P.K., 2017. Xssclassifier: An efficient xss attack detection approach based on machine learning classifier on snss. Journal of Information Processing Systems 13.

Riesthuis, P., Otgaar, H., Bücken, C., 2025. Ready to roc? a tutorial on simulation-based power analyses for null hypothesis significance, minimum-effect, and equivalence testing for roc curve analyses. Behavior Research Methods 57, 120.

Sant, L., 2025. Roc curves, in: International Encyclopedia of Statistical Science. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 2212–2215.

Santithanmanan, K., Kirimasthong, K., Boongoen, T., 2023. Machine learning based xss attacks detection method, in: UK Workshop on Computational Intelligence, Springer Nature Switzerland, Cham. pp. 418–429.

Shaisob, M.H., et al., 2025. Xss-safenet: A bidirectional lstm architecture for high-precision cross-site scripting detection, in: 2025 28th International Conference on Computer and Information Technology (ICCIT), IEEE.

Thajeel, I.K., Samsudin, K., Hashim, S.J., Hashim, F., 2023. Machine and deep learning-based xss detection approaches: a systematic literature review. Journal of King Saud University - Computer and Information Sciences 35, 101628.

Wali, S., Farrukh, Y.A., Khan, I., 2025. Explainable ai and random forest based reliable intrusion detection system. Computers & Security 157, 104542.

Younas, F., Raza, A., Thalji, N., Abualigah, L., Zitar, R.A., Jia, H., 2024. An efficient artificial intelligence approach for early detection of cross-site scripting attacks. Decision Analytics Journal 11, 100466.

Zeng, G., 2025. Invariance properties and evaluation metrics derived from the confusion matrix in multiclass classification. Mathematics 13, 2609.