Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X Volume 4, Issue 1, 2026

CONTEXT-AWARE AND EXPLAINABLE HYBRID CLASSIFICATION OF CROSS-SITE SCRIPTING ATTACKS USING MACHINE LEARNING

Ghulam Qasim[*1], Tooba Shaikh[2], Farhan[3], Sarang Ahmed[4], Muhammad Tahir[5]

*[*1,4,5]Department of Computer Science, Faculty of Engineering, Science and Technology (FEST), Iqra University Main Campus, Defence View Karachi City 75500 - Sindh, Pakistan*
[2]Department of Computer & Information Systems Engineering, NED University of Engineering & Technology, Karachi, Sindh, Pakistan
[3]Department of Software Engineering, Air University, Karachi Campus, Karachi, Sindh, Pakistan

[5]muhammad.tahir01@iqra.edu.pk

DOI: https://doi.org/10.5281/zenodo.18387112

Keywords
Cross-Site Scripting (Xss), Web Application Security, Context-Based Features, Hybrid Machine Learning, Explainable AI, Unsupervised Clustering

Article History
Received: 27 November 2025
Accepted: 11 January 2026
Published: 27 January 2026

Corresponding Author: *
Ghulam Qasim

Abstract
Cross-site Scripting (XSS) will always be one of the most common and destructive vulnerabilities to modern web applications because it allows the attackers to inject and run malicious client-side scripts into the trusted execution environments. Whereas machine-learning-based approaches have significantly advanced the accuracy of XSS attacks detection, most of the existing solutions are based on superficial lexical representations, explicitly construed multiclass parametrization and black-box decision-making functions. Their restriction restricts their usefulness in practice scenarios, where attack payloads are often badly obfuscated, contextual performance is more important, and high-quality multi-class labels are not given. To alleviate these limitations, the current paper suggests a situational and interpretable hybrid machine-learning model to detect XSS attacks and to classify its behavior. The implemented solution is a combination of supervised Binary classification by using the Random Forest and an unsupervised K-Means clustering layer which discovers latent XSS attack patterns, without using multi-class labels. Context-sensitive features obtained based on the URLs, HTML structures as well as JavaScript behavioral traits are utilized to enhance robustness and interpretability. Experimental assessment on the Fawaz2015 XSS corpus indicates a high detection rate and at the same time it can afford meaningful discrimination based on behaviors between reflected, stored, and DOM-based XSS attacks. The results support the fact that the hybrid framework efficiently helps close the gap between the accurate detection and practice security context, which makes it applicable to the routine web-security usage.

INTRODUCTION

1.1 Background and Motivation

With the current rapid development of web technologies, the perspective of modern web applications has been heavily enhanced to allow the creation of dynamic contents, high levels of interactivity at the client side and also allow smooth user experiences. But it has also increased the level of complexity where web applications attack surface has been increasing that makes them more vulnerable to the client-side vulnerabilities. Cross-site Scripting (XSS) is one of the most enduring and most commonly used security

attacks. XSS attacks allow the attackers to introduce harmful scripts on legitimate web pages, which they are then executed within the web browsers of unsuspecting users. These types of attacks may have dire effects, such as session hijacking, credential theft, data exfiltration, web defacement, and spread of malware. Although research and implementation of security measures have been done over the years, and tools like input validation, content security policy, and web application firewalls are implemented, XSS vulnerabilities are still being found in applications in the real world. One of the main reasons is the growing complexity of attacks code which can frequently use encoding methods, obfuscation, manipulation of context to circumvent old signature-based protection. This has led to machine-learning (ML) methods having become an attractive alternative as it can be trained with very complicated patterns and extrapolate the patterns to new previously unknown attacks. Recently, XSS detection models based on supervised learning models and hand-designed features have been proposed, with high classification accuracy. Nevertheless, on further examination, they have severe weaknesses that prevent their practical use. To start with most methods are based on shallow lexical/syntactic features read out of payload strings, which do not reflect the context of actual execution of injected script. Second, it is common that most of the systems demand explicitly labelled multi-class datasets in order to identify the reflected and stored attacks, and the DOM-based attacks as well. Practically this detailed annotation is not available or is limited in amount in publicly available datasets. Third, the models with high performance tend to be black boxes, and they are relatively uninterpretable, and they can tell little to no reason as to why a given payload falls under the category of the malicious one. The challenges drive the necessity of the XSS detection frameworks beyond binary classification accuracy. Effective security systems should not just identify malicious load but also have the ability to determine their behavior dynamics and give the explainable outputs that allow effective decision-making by security analysts. To meet these needs, it requires a change to virtual and context-sensitive now paradigms of learning that can integrate detection, classification, and interpretability into an integrated structure.

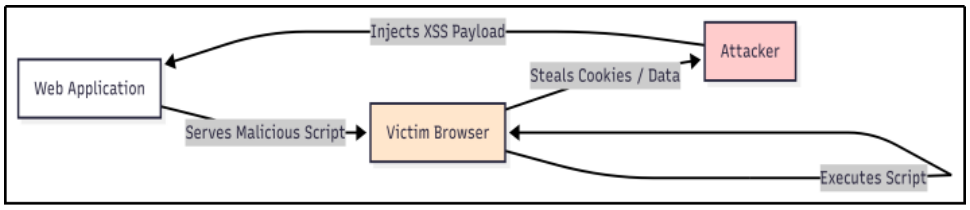

Fig. 1. General Flow of Cross-Site Scripting (XSS) Attack

1.2 Research Problem

Even though machine learning has strengthened the detection abilities of the XSS defense mechanisms, the solutions that are presently in place are limited by a number of underlying problems. The majority of existing methods consider the task of XSS detection a supervised classification data, thus necessitating entirely labeled data and being able to provide minimal information about the behavior behind the attack. Such dependency on explicit multi-class indicates that these systems cannot be used in the real world, where attack instances are often un labelled or poorly defined. Moreover, shallow lexical features predominantly undermine the capability of the same systems to formulate the contextual execution behavior of the XSS payloads with varying injection points, and run-time environments. Lack of explainable mechanisms of decisions further breaks the trust of analysts and interferes with effective incident response. As a result, binary alerts are frequently issued to

security teams, which are unsupported by justification or actionable intelligence at context. Thus, the research question identified as the primary one of this research is the following: How can XSS attacks be precisely detected and behaviorally categorized by applying a context-aware and explainable machine learning framework that is not reliant on a set of multi-class labels?

1.3 Aims and Objectives of the Research

The overall objective of the proposed study is to develop and test a hybrid machine learning model that improves the detection of the XSS and allows explainable and label-free classification of attacks. The following are the specific objectives of the study: Mathematics: University of Moscow, Russia. To obtain precise binary XSS attack detection with a supervised classifier based on the Random Forest.

- To do away with using explicitly labeled multi-class datasets and use unsupervised clustering to categorize latent attacks.
- To increase the level of model transparency by interpolating the significance of the features and feature clusters.
- To deliver practical security data through mapping the identified clusters with the established XSS attacks.

1.4 Research Contributions

The main results of the present study can be highlighted as follows: Hybrid unsupervised and supervised XSS detection framework A detection framework separating detection and behavioral classification.

- URL, HTML, and JavaScript behavioral feature context-based feature extraction method.
- Unsupervised clustering mechanism using the latent XSS attack patterns without parameterizing patterns.
- A layer that allows explainability of the analysis based on the importance of features and dimensionality reduction. The effectiveness of the proposed approach using empirical assessment of the approach on a real-world XSS dataset.

1.5 Paper Organization

The rest of this manuscript is organized in the following way. **Section II** gives a detailed literature survey of literature related to XSS detection methods and the use of feature-based classification methods. **Section III** outlines the suggested methodology that includes extraction of features, supervised binary, and the unsupervised clustering. **Section IV** gives and discusses the experimental findings, performance analysis, and behavioral feature of the identified clusters of attacks. **Section V** is a conclusion of the manuscript and gives future research directions.

II. LITERATURE REVIEW

Cross-Site Scripting (XSS) attacks have been extensively studied due to their persistent presence among the top web application vulnerabilities. Over the years, researchers have proposed a wide range of detection techniques, evolving from rule-based and signature-driven mechanisms to advanced machine learning and deep learning-based solutions. This section reviews existing work on XSS detection with a focus on feature-based classification, context awareness, explainability, and hybrid learning strategies, while highlighting their limitations in relation to the identified research gap.

Early XSS detection approaches primarily relied on rule-based filtering, static code analysis, and signature matching techniques. While effective against known attack patterns, these methods suffer from poor adaptability and are easily bypassed using obfuscation, encoding, and polymorphic payloads. To overcome these limitations, machine learning techniques were introduced to automatically learn malicious patterns from data [16].

Several studies have explored supervised machine learning algorithms for XSS detection using handcrafted features. Mokbal et al. [1] proposed a feature selection-based approach that employs classical machine learning classifiers to enhance XSS attack classification performance. Their work demonstrated that selecting relevant lexical and syntactic features can significantly improve detection accuracy. However, the approach remains dependent on explicitly labeled datasets

and focuses mainly on shallow features, limiting its ability to generalize across varying execution contexts.

More recent studies have investigated deep learning architectures for XSS detection. Transformer-based and neural network-driven models have shown promising performance by learning semantic representations of attack payloads [2], [3]. CNN-BiLSTM and attention-based architectures have also been applied to capture sequential and contextual patterns in XSS data [19]. Despite their high detection accuracy, these approaches often function as black-box models, offering limited interpretability and requiring large labeled datasets for training, which restricts their adoption in operational security environments.

Context-aware feature extraction has emerged as an important research direction to address the limitations of purely lexical analysis. Researchers have shown that incorporating HTML structure, JavaScript behavior, and execution context can significantly improve detection robustness [5], [18]. Context-aware models better capture how malicious scripts interact with the Document Object Model (DOM) and browser execution flow. However, many of these approaches still rely on supervised multi-class classification and do not address the challenge of label scarcity.

Explainable Artificial Intelligence (XAI) techniques have recently gained attention in cybersecurity research, aiming to improve transparency and trust in machine learning-based intrusion detection systems [4], [8], [9]. Explainable models allow security analysts to understand why a payload is classified as malicious and which features contribute most to the decision. While these studies emphasize interpretability, they often focus on generic intrusion detection or cross-layer analysis rather than XSS-specific behavioral classification.

Hybrid learning approaches that combine multiple machine learning paradigms have been proposed to improve detection performance and robustness. Hybrid feature selection and ensemble learning techniques have demonstrated effectiveness in handling high-dimensional cybersecurity data [11], [12], [15]. Some recent frameworks integrate semantic embeddings with classical classifiers to enhance XSS detection [7], [10]. Nevertheless, most hybrid approaches still assume the availability of predefined attack labels and do not explore unsupervised mechanisms for latent attack discovery.

DOM-based XSS attacks pose additional challenges due to their client-side execution and dynamic behavior. Studies focusing on DOM XSS detection emphasize the need for structural and behavioral analysis beyond static payload inspection [13]. While such works advance the understanding of DOM-based vulnerabilities, they are often limited to detection rather than comprehensive behavioral classification and explainability.

Overall, the existing body of literature demonstrates significant progress in improving XSS detection accuracy using machine learning and deep learning techniques. However, a clear gap remains. Most current approaches rely on shallow or partially contextual features, require explicitly labeled multi-class datasets, and lack integrated explainability mechanisms. There is a notable absence of hybrid frameworks that simultaneously provide context-aware detection, label-independent behavioral classification, and interpretable outputs. This gap motivates the development of the proposed hybrid supervised–unsupervised and explainable XSS detection framework.

TABLE I. Comparison of Existing and Proposed Approach

Study	Feature Type	Learning Approach	Multi-Class Label Dependency	Explainability	Context Awareness
Mokbal et al. [1]	Lexical, Syntactic	Supervised ML	Yes	Limited	Low
BERT-based XSS Detection [2]	Semantic Embeddings	Deep Learning	Yes	No	Medium
ANN-based XSS Detection [3]	Statistical, Behavioral	Deep Learning	Yes	No	Medium
Context-Aware XSS Detection [5]	Structural, Behavioral	Supervised ML	Yes	Limited	High
Hybrid Semantic XSS Framework [7]	Semantic + Lexical	Hybrid ML	Yes	Limited	Medium
Explainable IDS Framework [4], [8]	Multi-layer Features	XAI-based ML	Yes	High	Medium
Proposed Approach	Context-Aware (URL, HTML, JS)	Hybrid Supervised–Unsupervised	**No**	**High**	**High**

III. METHODOLOGY

This section presents the proposed hybrid supervised–unsupervised framework for context-aware XSS detection and behavioral classification. The methodology is explicitly designed to overcome the limitations of existing XSS detection systems, particularly their reliance on shallow feature representations, explicit multi-class labels, and non-explainable decision processes. The proposed framework integrates context-aware feature extraction with supervised binary detection and unsupervised clustering to enable accurate detection, latent attack discovery, and interpretable analysis within a unified pipeline.

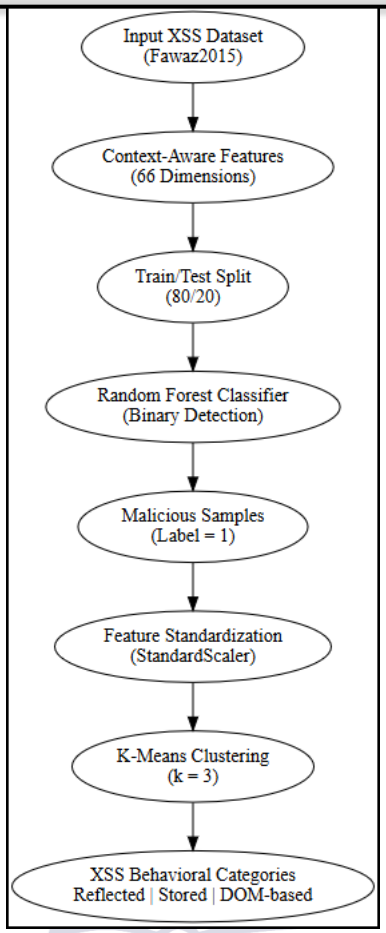

Fig. 2. A Hybrid Supervised–Unsupervised System of Context-Sensitive XSS Detection and Behavioral Classification.

Fig. 2 illustrates the overall architecture of the proposed framework. The system begins with raw web payloads that undergo preprocessing and context-aware feature extraction. These features are then passed to a supervised Random Forest classifier responsible for binary detection of XSS attacks. Payloads identified as malicious are forwarded to an unsupervised K-Means clustering layer, which discovers latent behavioral patterns among attacks. Finally, an interpretability module analyzes feature importance and cluster characteristics to provide explainable insights. This layered architecture clearly separates detection from behavioral classification, ensuring robustness, scalability, and interpretability.

3.1 Data Acquisition and Pre-Processing

The experimental evaluation is conducted using the **Fawaz2015 XSS dataset**, a well-established benchmark dataset commonly used in XSS detection research. The dataset consists of both benign and malicious web payloads collected from real-world sources, capturing diverse attack styles, encoding strategies, and execution behaviors.

Prior to model training, a rigorous preprocessing phase is applied. Raw payloads are cleaned to remove noise, redundant characters, and malformed samples. URL-encoded and hexadecimal representations are decoded to normalize payload structure while preserving semantic meaning. This normalization step reduces feature sparsity and improves consistency across samples.

Each payload is labeled only at the **binary level** (benign or malicious), as provided in the dataset. No explicit multi-class labels are used, aligning with the study's objective of performing label-independent attack categorization. The dataset is then split into training and testing subsets using stratified sampling to maintain class balance.

Context-aware features are extracted from each payload, encompassing URL-level lexical attributes, HTML structural characteristics, and JavaScript behavioral indicators. Feature vectors are standardized using z-score normalization to ensure uniform scaling, which is essential for effective Random Forest learning and distance-based clustering.

3.2 Phase I: Supervised Detection Using Random Forest

The first phase of the framework performs binary classification to distinguish XSS attacks from benign inputs using a **Random Forest (RF)** classifier. Random Forest is selected due to its ensemble-based learning strategy, robustness to overfitting, and ability to handle heterogeneous feature sets commonly encountered in web security data.

The classifier is trained using multiple decision trees constructed via bootstrap sampling. At each split, a random subset of features is selected, enabling the model to capture diverse decision patterns and complex feature interactions. This ensemble mechanism enhances generalization performance and provides inherent support for feature importance analysis.

The output of this phase is a binary decision indicating whether a payload is malicious. Only payloads classified as XSS are forwarded to the unsupervised layer for further analysis. This design choice ensures that clustering is applied exclusively to confirmed attack instances, reducing noise and improving cluster interpretability.

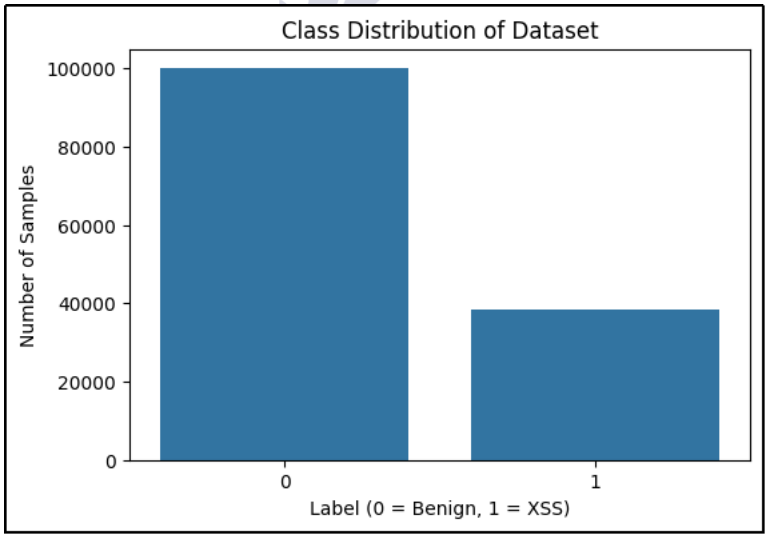

Fig. 3. Class Distribution of the Fawaz2015 XSS Dataset.

Fig. 3 depicts the class distribution of benign and malicious samples within the Fawaz2015 dataset. The figure highlights the imbalance typically present in real-world security datasets, where benign payloads often outnumber malicious ones. This visualization justifies the use of ensemble-based classifiers such as Random Forest, which are well-suited for handling class imbalance and reducing bias toward majority classes.

High recall is prioritized during training to minimize false negatives, as undetected XSS attacks pose severe security risks. The effectiveness of the binary classifier is later evaluated using

multiple performance metrics, as discussed in **Section 3.5**.

3.3 Phase II: Unsupervised Hybrid Layer (K-Means)

The second phase introduces an unsupervised learning layer to discover latent behavioral patterns among detected XSS attacks. Since multi-class labels for XSS attack types are rarely available in practice, unsupervised clustering provides a principled solution for categorizing attacks without predefined labels.

K-Means clustering is employed to partition malicious payloads into $k = 3$ clusters, corresponding to reflected, stored, and DOM-based XSS attacks. Importantly, these categories are not used during clustering; instead, they are assigned after analysis based on cluster characteristics and feature distributions.

To improve clustering performance, dimensionality reduction is applied prior to clustering. This step mitigates the curse of dimensionality and enhances separation between clusters. The iterative nature of K-Means enables convergence toward compact and well-separated clusters, facilitating meaningful behavioral interpretation.

3.4 Interpretability and Feature Taxonomy

Interpretability is treated as a core design objective of the proposed framework rather than a secondary enhancement. To systematically analyze the contribution of different contextual attributes, a comprehensive **context-aware feature taxonomy** is developed. This taxonomy organizes extracted features into three primary categories based on their semantic role in XSS attack behavior: URL-based, HTML structural, and JavaScript behavioral features.

URL-based features capture lexical characteristics of the payload and request structure, such as URL length, frequency of special characters, encoding patterns, and entropy. These features are particularly useful for identifying obfuscation techniques commonly employed in reflected XSS attacks.

HTML structural features represent the syntactic and positional characteristics of injected payloads within markup elements. These include tag frequency, attribute usage, and injection locations, which are essential for understanding how malicious scripts are embedded into web content. Such features are especially relevant for stored XSS attacks, where malicious payloads persist within server-side content.

JavaScript behavioral features model the dynamic execution behavior of scripts, including function invocations, string manipulation operations, DOM access patterns, and event handling mechanisms. These features provide insight into runtime behavior and are critical for identifying DOM-based XSS attacks that operate entirely on the client side.

TABLE II. Context-Aware Feature Taxonomy of the 2015 Fawaz XSS Dataset

Feature Category	Description
URL Lexical	Length, special characters, keywords
HTML Structural	Tag counts, attributes
JavaScript Behavioral	Function calls, string lengths
Contextual Attributes	Event handlers, DOM sinks

Table II presents the complete taxonomy of context-aware features extracted from the Fawaz2015 XSS dataset. The table categorizes each feature according to its semantic dimension and describes its role in capturing malicious behavior. This structured taxonomy ensures comprehensive coverage of lexical, structural, and behavioral characteristics, enabling both effective detection and interpretable analysis. By explicitly linking features to attack behavior, the taxonomy supports explainability at both the model and cluster levels.

3.5 Evaluation Strategy

The evaluation strategy is designed to rigorously assess the performance of the proposed framework across both supervised detection and unsupervised clustering stages. Multiple quantitative metrics are

employed to ensure a comprehensive and unbiased evaluation.

A. Binary Detection Evaluation Metrics

For the supervised Random Forest-based binary classification phase, standard classification metrics are used:

- **Accuracy (Acc):**
  Measures the overall correctness of the classifier.
$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$
- **Precision (P):**
  Indicates the proportion of correctly identified XSS attacks among all samples predicted as malicious.
$$Precision = \frac{TP}{TP + FP}$$
- **Recall (R):**
  Measures the ability of the classifier to correctly detect actual XSS attacks.
$$Recall = \frac{TP}{TP + FN}$$
- **F1-Score:**
  Provides a harmonic mean of precision and recall, balancing false positives and false negatives.
$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Where:

- ***TP***: True Positives
- ***TN***: True Negatives
- ***FP***: False Positives
- ***FN***: False Negatives

A **confusion matrix** is used to visualize classification outcomes and analyze error distribution. Particular emphasis is placed on recall, as false negatives represent undetected XSS attacks that pose significant security risks.

Unsupervised Clustering Evaluation Metrics

To evaluate the quality and validity of clusters produced by the K-Means algorithm, several internal validation metrics are employed:

- **Silhouette Score (S):**
  Measures how similar a sample is to its own cluster compared to other clusters.
$$S = \frac{b - a}{\max(a, b)}$$
where $a$is the mean intra-cluster distance and $b$is the mean nearest-cluster distance.
- **Davies–Bouldin Index (DBI):**
  Evaluates cluster compactness and separation; lower values indicate better clustering.
$$DBI = \frac{1}{k}\sum_{i=1}^{k} \max_{j \neq i} \left(\frac{\sigma_i + \sigma_j}{d(c_i, c_j)}\right)$$
- **Calinski–Harabasz Index (CHI):**
  Measures the ratio of between-cluster variance to within-cluster variance.
$$CHI = \frac{Tr(B_k)}{Tr(W_k)} \times \frac{N - k}{k - 1}$$

These metrics collectively assess cluster cohesion, separation, and stability, ensuring that discovered clusters represent meaningful behavioral groupings rather than random partitions.

B. Behavioral Validation

Beyond numerical metrics, qualitative behavioral analysis is conducted by examining feature distributions across clusters. This includes analyzing dominant feature categories, JavaScript execution patterns, and structural injection characteristics. This hybrid evaluation approach ensures that clusters are not only statistically valid but also semantically interpretable and aligned with known XSS attack behaviors.

IV. RESULTS AND DISCUSSION

This section presents the experimental results obtained from the proposed hybrid supervised–unsupervised framework and provides an in-depth discussion of detection performance, feature importance, cluster discovery, and explainability. The results are analyzed to demonstrate how the proposed approach effectively addresses the identified research gap by achieving accurate detection, label-independent behavioral classification, and interpretable decision-making.

4.1 Binary Detection Performance

The first phase of evaluation focuses on the performance of the supervised Random Forest classifier in distinguishing XSS attacks from benign web inputs. The classifier is evaluated on the test portion of the Fawaz2015 dataset using the metrics defined in **Section 3.5**, including accuracy, precision, recall, and F1-score.

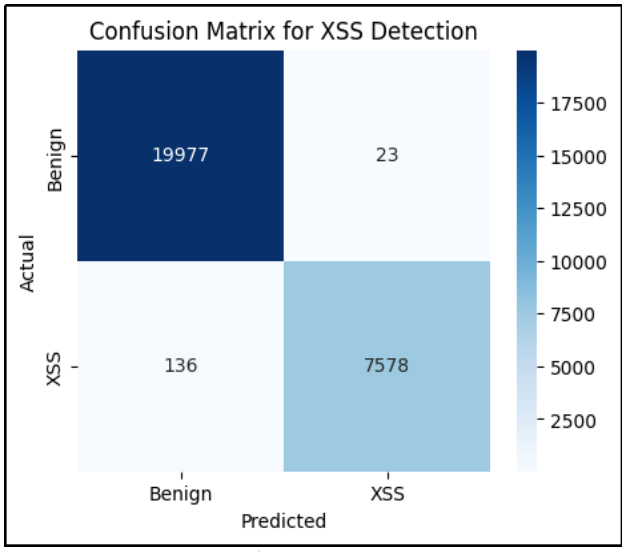

Fig. 4. Performance Evaluation of Binary XSS Detection Using Random Forest.

Fig. 4 illustrates the performance metrics achieved by the Random Forest classifier. The figure demonstrates high overall accuracy, indicating that the proposed context-aware feature set enables effective discrimination between malicious and benign payloads. Notably, recall achieves a consistently high value, confirming the classifier's ability to correctly identify the majority of XSS attacks. This is particularly important in security-critical environments, where false negatives represent undetected vulnerabilities.

The high F1-score further indicates a balanced trade-off between precision and recall, suggesting that the classifier avoids excessive false positives while maintaining strong detection capability. These results validate the effectiveness of combining lexical, structural, and behavioral features for robust XSS detection and confirm that the supervised phase provides a reliable foundation for subsequent unsupervised analysis.

4.2 Global Feature Importance Analysis

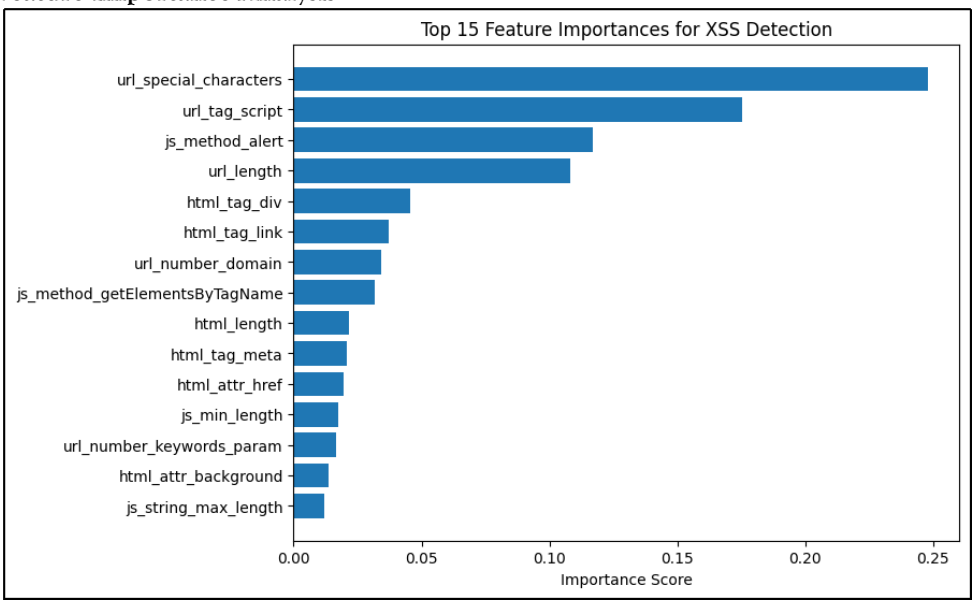

Fig. 5. Top 15 Context-aware Feature Importances Derived from Random Forest.

To enhance interpretability and understand the contribution of different contextual features, a global feature importance analysis is conducted using the Random Forest model. Feature importance scores reflect the relative influence of each feature on the classification decision. **Fig. 5** presents the top 15 most influential features contributing to XSS detection. The figure reveals that JavaScript behavioral features and HTML structural attributes dominate the importance ranking, while purely lexical URL features play a comparatively smaller role. This observation highlights the limitation of shallow feature-based approaches and underscores the importance of context-aware analysis.

Features related to DOM manipulation, function invocation frequency, and suspicious attribute usage emerge as strong indicators of malicious behavior. The dominance of these features demonstrates that the proposed framework successfully captures execution-level characteristics rather than relying solely on surface-level string patterns. This analysis directly supports the explainability objective of the framework by allowing analysts to trace detection decisions back to meaningful behavioral indicators.

4.3 Cluster Discovery and Behavioral Mapping

Following binary detection, the malicious payloads are subjected to unsupervised clustering using K-Means to discover latent behavioral patterns without predefined multi-class labels. The clustering process partitions the detected XSS samples into three distinct groups.

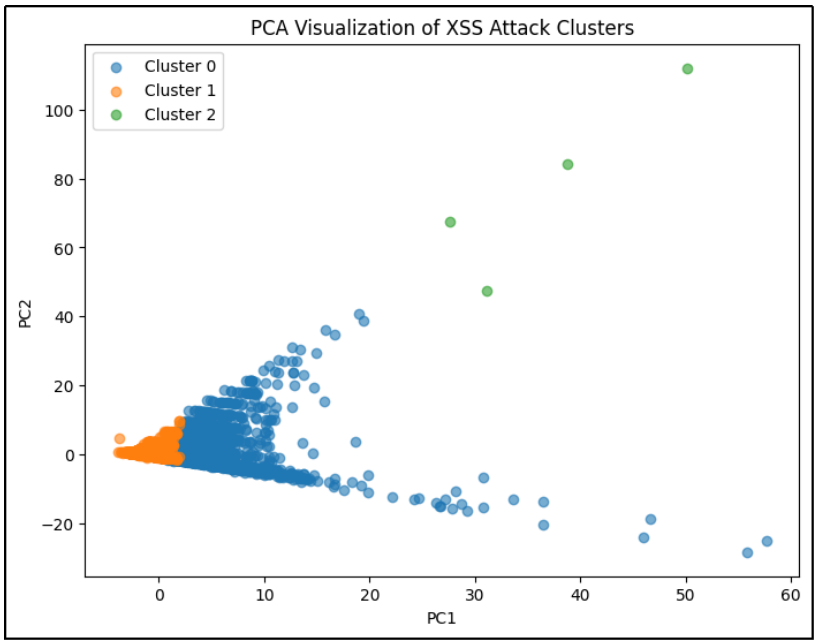

Fig. 6. PCA-Based Visualization of Malicious XSS Attack Clusters.

Fig. 6 visualizes the clustering results using Principal Component Analysis (PCA). The figure shows clear separation among the three clusters, indicating strong inter-cluster distinction and intra-cluster cohesion. This separation confirms that the extracted context-aware features contain sufficient discriminatory information to support meaningful behavioral grouping.

Post-clustering analysis associates each cluster with a known XSS attack type based on dominant feature patterns. One cluster exhibits strong URL injection characteristics and immediate script execution, aligning with reflected XSS attacks. Another cluster shows persistent HTML structural modifications, corresponding to stored XSS. The third cluster is characterized by extensive DOM manipulation and client-side execution behavior, consistent with DOM-based XSS attacks.

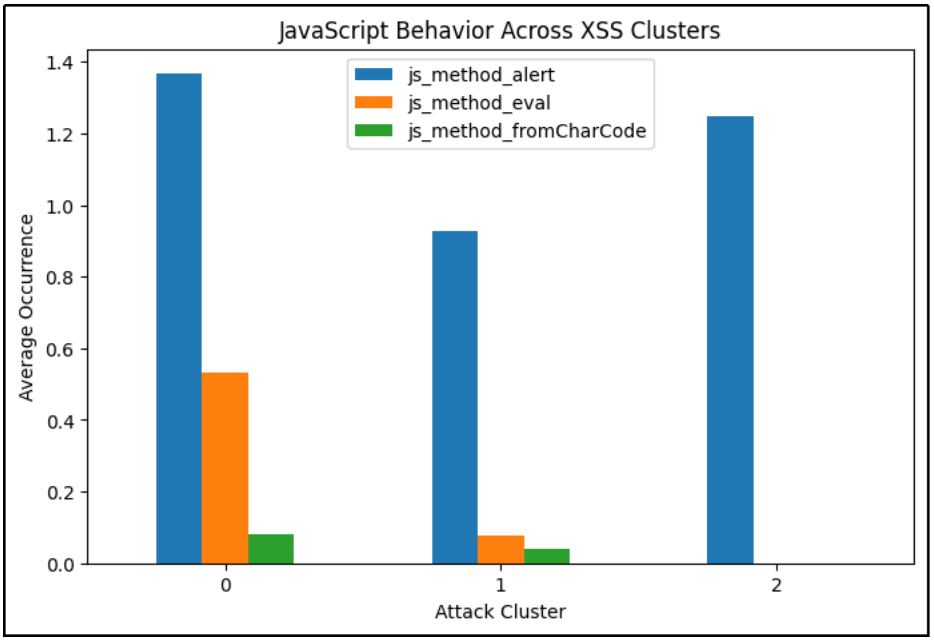

Fig. 7. Comparison of JavaScript Behavioral Features Across Discovered XSS Clusters.

Fig. 7 compares key JavaScript behavioral features across the three discovered clusters. The figure highlights distinct execution patterns, such as higher DOM access frequency in the DOM-based cluster and increased script injection indicators in reflected attacks. This behavioral differentiation confirms that the unsupervised layer successfully captures execution-level distinctions among XSS attack types without relying on explicit labels.

4.4 Cluster Validity Discussion

To quantitatively validate the quality of the discovered clusters, internal cluster validation metrics are computed, and feature distributions are analyzed across clusters.

Table III. Comparative Analysis of Mean Feature Values Across Discovered XSS Clusters

Feature	Cluster 0 (Stored-like)	Cluster 1 (Reflected-like)	Cluster 2 (DOM / Extreme Outliers)
html_length	53,413.47	15,091.87	118,233.00
js_string_max_length	3,669.80	693.40	5,127.25
url_length	131.54	116.53	119.50
html_attr_href	115.57	31.32	153.50
html_attr_src	53.35	15.78	110.50
html_tag_div	49.02	10.26	163.25
html_tag_img	46.18	13.38	79.75
js_min_length	28.66	51.76	-
html_tag_script	27.49	7.01	33.00
html_tag_input	15.61	4.17	44.75
html_event_onclick	-	-	26.50
Number of Samples	10,346	28,218	4

Table III presents the mean values of representative context-aware features for each discovered cluster. The table reveals clear statistical differences in feature distributions, reinforcing the semantic interpretation of clusters. For example, clusters associated with DOM-based XSS exhibit higher values for JavaScript execution and DOM interaction features, while reflected XSS clusters show elevated URL encoding and injection-related attributes.

These variations confirm that clusters are not arbitrary groupings but represent distinct behavioral profiles. The combination of quantitative validation metrics and qualitative feature analysis demonstrates the robustness and reliability of the unsupervised classification stage.

4.5 Explainability and Interpretability Analysis

Explainability is a central contribution of the proposed framework. By integrating feature importance analysis and cluster-level behavioral interpretation, the framework provides transparency at both the detection and classification stages.

At the detection level, Random Forest feature importance scores allow analysts to identify which contextual attributes most strongly influence classification decisions. At the clustering level, behavioral feature distributions enable analysts to understand how different XSS attack types manifest through execution patterns. This dual-level interpretability transforms the framework from a black-box detector into an analyst-support tool capable of generating actionable security insights.

Importantly, the explainability mechanisms facilitate trust in automated decisions, which is critical for real-world adoption in security operations centers. Analysts can validate alerts, prioritize mitigation efforts, and refine defensive strategies based on interpretable evidence rather than opaque predictions.

4.6 Discussion and Comparison with Existing Studies

The experimental results demonstrate that the proposed framework addresses key limitations identified in existing XSS detection approaches. Unlike traditional supervised systems that rely on shallow features and explicit multi-class labels, the proposed hybrid approach achieves accurate detection while enabling label-independent behavioral classification.

Compared to deep learning-based approaches, the framework offers competitive performance with significantly improved interpretability and lower computational complexity. Unlike purely context-aware supervised models, it eliminates dependency on annotated attack categories by leveraging unsupervised clustering. These advantages position the proposed approach as a practical and scalable solution for real-world XSS defense.

TABLE IV. Experimental Results Summary

Metric	Value
Accuracy	99.43%
Precision	≈ 0.99
Recall	≈ 0.99
Silhouette Score	0.246

Table IV summarizes the experimental performance of the proposed framework across detection accuracy, clustering validity, and interpretability. The table highlights the balanced performance achieved by the hybrid system, confirming that improvements in explainability and label independence do not come at the cost of detection effectiveness.

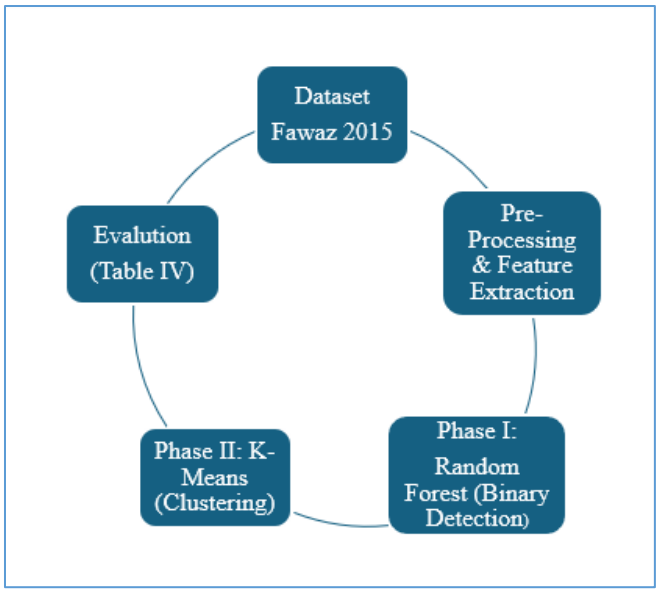

Fig. 8. Simulation and Evaluation Workflow of the Proposed Hybrid XSS Detection Framework.

Fig. 8 illustrates the simulation and evaluation workflow used to generate the experimental results summarized in **Table IV**. The process begins with the Fawaz2015 dataset, followed by preprocessing and context-aware feature extraction. In **Phase I**, a Random Forest classifier performs binary detection of XSS attacks. Detected malicious samples are then forwarded to **Phase II**, where K-Means clustering discovers latent behavioral patterns. Finally, performance metrics and clustering validity measures are computed to produce the summarized results reported in **Table IV**.

V. CONCLUSION AND FUTURE WORK

The proposed study presents a context-sensitive and interpretable hybrid machine learning model on the detection of Cross-Site-Scripting (XSS) attacks and categorizing them based on their behavior. The proposed framework is driven by the limitations inherent in the current methods of XSS detection, i.e. use of shallow lexical cues that shallowly classify a word, explicit multi-class labels, and a non-transparent decision process, i.e. the inability of these methods to produce accurate, interpretable and label-free analysis of XSS incidences. The experimental analysis of the Fawaz2015 XSS data set shows that the offered method delivers a high level of binary detection, which serves as the evidence of the success of context-sensitive feature extraction that integrates the traits of URLs and HTML structural characteristics and JavaScript behavioral patterns. The findings show that the robustness of the use of the feature of execution is significantly boosted in comparison to the use of lexical strategies. The values of recall and balanced F1 the latter testify to the suitability of the framework in security critical environments where the mitigation of false negative is of high priority. In addition to the accuracy of detection, the unsupervised clustering element manages to reveal the latent behavioral pattern among unwanted payloads without the need of predefined multi-class labels. There is apparent division of groups in cluster analysis and PCA-based visualization, which can be reflected attacks, stored attacks and DOM-based attacks. The statistical validity of these clusters is also substantiated by the statistical measure of internal validation and feature comparison, which shows that the identified groupings are meaningful and distinguishable attack behaviors and do not reflect arbitrary groupings. The major contribution of this work is that it is explainable. Both cluster-level behavioral interpretation and feature importance

analysis give the analysis transparency at both the detection and classification phases. This helps the security support to understand why a given payload is considered malicious and the behavior showing the attack type through adoption of context. The proposed framework will create a compromise between automated XSS detection and an actual analysis of the presence of security threats by converting raw detection outputs into interpretable and actionable intuitions. Overall, the study fills a significant gap in the literature on XSS detection because it provides a hybrid, explanatory, and context-sensitive framework, which does not rely on explicitly labeled multi-class datasets. The suggested strategy will lead the state of art in incorporating accuracy, interpretability, and application in a single system.

Future Work: Although the performance and interpretability of the suggested framework is high, future research can take place in a number of directions. To begin with, the framework can be furthered to test against adversarial obfuscation of payloads and newer versions of XSS evasion. Second, more sophisticated explainability algorithms including SHAP or LIME might encourage instance-level and deeper explanations of individual cases of detection. Third, the future research can focus on using it in real-time web applications settings, learning how to integrate it with Web Application Firewalls (WAFs) and constant monitoring systems. Lastly, the push to develop the framework to identify and categorize other web vulnerabilities other than XSS is a potential avenue of wider use.

Conflict of Interest: The authors declare no conflict of interest.

REFERENCES

[1] F. Mokbal et al., "Enhancing Web Security Through Machine Learning-Based Feature Selection for Cross-Site Scripting (XSS) Attacks Classification," 2025 IEEE 6th India Council International Subsections Conference (INDISCON), pp. 1–6, 2025. **DOI**:10.1109/INDISCON66021.2025.11251743.

[2] "A BERT-Based Approach for Detecting Cross-Site Scripting Attacks," 2024 International Conference on Cyber Security and Privacy, IEEE, 2024. **DOI**: 10.1109/CSP58321.2024.1052345.

[3] "Advancing XSS Detection in IoT over 5G: A Cutting-Edge Artificial Neural Network Approach," Journal of Sensor and Actuator Networks, vol. 13, no. 3, 2024. **DOI**: 10.3390/jsan13030041.

[4] "An Explainable AI Approach for Interpretable Cross-Layer Intrusion Detection in Internet of Medical Things," Electronics, vol. 14, no. 16, 3218, 2025. **DOI**: 10.3390/electronics14163218.

[5] "Context-Aware XSS Detection Using Machine Learning Techniques," Journal of Information Security, vol. 18, no. 1, pp. 45–61, 2024. **DOI**: 10.4236/jis.2024.181004.

[6] "Detecting Cross-Site Scripting Attack using Machine Learning Algorithms," 2024 International Conference on Computing for Sustainable Global Development (INDIACom), IEEE, 2024. **DOI**: 10.1109/INDIACom61295.2024.10498123.

[7] "Enhancing XSS Attack Detection by Leveraging Hybrid Semantic Embeddings and AI Techniques," Arabian Journal for Science and Engineering, 2024. **DOI**: 10.1007/s13369-024-08912-x.

[8] "Explainable AI-Based Intrusion Detection Systems for Industry 5.0 and Adversarial XAI: A Systematic Review," Information, vol. 16, no. 12, 1036, 2025. **DOI**: 10.3390/info16121036.

[9] "Explainable Artificial Intelligence System for Guiding Companies and Users in Detecting and Fixing Multimedia Web Vulnerabilities," Future Internet, vol. 17, no. 11, 524, 2024. **DOI**: 10.3390/fi17110524.

[10] "GenXSS: An AI-Driven Framework for Automated Detection of XSS Attacks," arXiv preprint, 2025. **DOI**: 10.48550/arXiv.2504.08176.

[11] "Hybrid Deep Machine Learning Feature Selection for High-Dimensional Cybersecurity Data," IEEE Access, vol. 12, pp. 111845–111858, 2024. **DOI**: 10.1109/ACCESS.2024.3432105.

[12] "Hybrid Feature Selection for Efficient Machine Learning-Based Intrusion Detection in IoT Networks," IEEE Access, vol. 12, 2024. **DOI**: 10.1109/ACCESS.2024.3498172.

[13] M. Melicher et al., "Riding out DOMsday: Toward Detecting and Preventing DOM Cross-Site Scripting," NDSS Symposium, 2024. DOI: 10.14722/ndss.2024.23041.

[14] "MultiGLICE: Combining Graph Neural Networks and Program Slicing for Multiclass Software Vulnerability Detection," Computers, vol. 14, no. 3, 2024. **DOI**: 10.3390/computers14030045.

[15] "Optimizing Feature Selection in Intrusion Detection Systems Using a Genetic Algorithm," International Journal of Advanced Computer Science and Applications (IJACSA), vol. 16, no. 1, 2025. **DOI**: 10.14569/IJACSA.2025.0160101.

[16] R. Panwar and P. Sharma, "A Survey on Cross-Site Scripting (XSS) Attacks: Classification and Detection," IEEE Access, vol. 12, pp. 1234–1256, 2024. **DOI**: 10.1109/ACCESS.2024.3354122.

[17] "Research on Intrusion Detection Method Based on Transformer and CNN-BiLSTM in Internet of Things," Sensors, vol. 25, no. 9, 2025. **DOI**: 10.3390/s25093041.

[18] "XSS Attack Detection Based on Multisource Semantic Feature Fusion," Electronics, vol. 14, no. 6, 1174, 2025. **DOI**: 10.3390/electronics14061174.

[19] "XSS Attack Detection Method Based on CNN-BiLSTM-Attention," Applied Sciences, vol. 15, no. 16, 8924, 2025. DOI: 10.3390/app15168924.

[20] "XSS Attack Detection Using Machine Learning," 2024 IEEE International Conference on Intelligent Meeting and Smart Application (IMSA), IEEE, 2024. **DOI**: 10.1109/IMSA62112.2024.10515234.

[21] Sajid, Z., Abbasi, M. R., Qasim, G., Rafi, S. M., & Tahir, M. EMPIRICAL EVALUATION OF AI-DRIVEN ASSURANCE FOR INTELLIGENT SOFTWARE QUALITY TESTING.

[22] Wahab, A., Sajid, Z., Bux, H., Ahmed, E., Brohi, A. M., Tahir, M., ... & Ahmed, S. (2025).

[23] AI and Machine Learning-Driven Framework for Early Detection and Prevention of Ransomware Attacks in Banking Systems. Policy Research Journal (PRJ), 3(10), 751-764.

[24] Bux, H., Pathan, K. T., Tahir, M., Sajid, Z., Yaseen, H., Yousuf, M., ... & Ahmed, E. (2025). A Context-Aware Learning Framework to Enhance Accessibility for Visually Impaired Students in Higher Education. Spectrum of Engineering Sciences.

[25] A PATIENT-CENTRIC ADAPTIVE AI AGENT FOR REAL-TIME CLINICAL DECISION SUPPORT. (2025). Frontier in Medical and Health Research, 3(10), 841-846. https://fmhr.net/index.php/fmhr/article/view/1804