A Review on Cross Site Scripting (XSS) Attack

Mrinal Goswami[1], Sreya Bhowmick[1], Barasha Das[1], Liaren Emani Aier[1], and Arpita Nath Boruah[1,*]

ARTICLE INFO

Article history:
Received: 3 January 2024
Revised: 18 December 2024
Accepted: 1 February 2025
Online: 15 May 2026

Keywords:
Cross site scripting (XSS)
Cybersecurity
Malicious scripts
Webpages

ABSTRACT

Cybersecurity is crucial because it protects digital systems and data from a variety of threats. One such significant threat is Cross Site Scripting (XSS), in which hackers compromise user data and privacy via injected malicious scripts, exploiting web page security weaknesses. XSS began in the early 2000s, with incidents growing by 20% annually over the past three years, continuing to be a top cybersecurity threat according to OWASP. Q1 2023 saw a 22% surge in reported incidents from the previous year, with projected 2023 financial losses surpassing $9 billion due to successful XSS attacks. Notably, in 2023 the detection time for XSS incidents dropped to 35 days, highlighting better response abilities. Advanced machine learning reduced false positives by 17%, aiding XSS prevention. In 2023, a cyber initiative led to a 15% reduction in social engineering-based XSS attacks through user education. In this review, various XSS works published between 2010 to 2023 are studied, putting a particular emphasis on the researchers' methods for detecting XSS attacks. This paper also addresses the main causes and effects of XSS attacks, the current XSS detection mechanisms, tools, and the shortcomings of those techniques.

1. INTRODUCTION

Cyberattacks are malicious actions carried out online to jeopardize the privacy, availability, or integrity of computer systems, networks, and data [1]. The degree of sophistication and popularity of cyberattacks are constantly increasing with hackers not only targeting individuals but also organizations and governments. Because of all these reasons, cybersecurity is crucial.

The practice of defending computer systems and networks from cyberattacks is known as cybersecurity. Due to the prevalence of digital technology in today's society, cybersecurity is crucial to counter cyberattacks. Individual confidentiality, stability in finances, and even national security are at risk in the absence of effective cybersecurity measures. It delivers the tactics and defense systems to safeguard our digital assets and plays a part in protecting the accuracy of data and digital trust. Sources say that every 3.17 seconds, a brand-new malware program is created. In [2] emphasizes the significance of cyberspace as a source of power and discusses how power extends beyond governments to private corporations, terrorists, and individuals in the digital age. It underscores the necessity of collaboration between public and private sectors to address location-independent and highly destructive cyber threats. As our dependency on technology increases, the relevance of cybersecurity will only grow with it, and recognizing the need for investment in this area will protect our digital future [3][4].

One such cyber threat is XSS [6]. The term "XSS" was initially coined by a Microsoft security engineer in January 2000. XSS is a dangerous flaw in web-based applications that enables attackers to inject malicious scripts into trustworthy websites, potentially impacting the users who use those sites. Any vulnerable website can be subjected to an XSS attack, irrespective of the programming language used to create it. XSS attacks tend to be paired with other types of attacks, like phishing and social engineering attacks. The Gmail attack in 2007, the Twitter attack in 2009, and the Yahoo attack in 2013 are some of the most popular XSS attacks.

XSS attacks can be highly harmful since they give hackers the ability to steal confidential data, take over account details, or even spread malware. Cookies can be stolen during XSS attacks to impersonate the victim for hacking their accounts [7] [8]. Attackers can distribute spam or malicious links to contacts using the accounts of the hacked user. User privacy can be violated via XSS attacks that expose private data which can lead to victims experiencing psychological distress. Hence, an XSS attack may result in a lifelong negative impact on one's finances, emotions, and career. Regardless of the many methods developed by dedicated researchers for the detection and

[1]*Programme of CSE, Faculty of Computer Technology, Assam down town University, Panikhaiti, Guwahati, India.*
**Corresponding author:* Arpita Nath Boruah; Email: arpita.b@adtu.in.

prevention of these vulnerabilities, web pages can still be severely affected by XSS attacks. Detection of XSS attacks remains challenging due to a variety of attack options, the difficulty in differentiating malicious scripts from authentic ones, and continuously stable browser behavior. Figure 1 gives a general illustration of XSS attack.

The rest of the paper is structured as follows: Section 2 discusses the different categories of XSS attacks, Section 3 talks about XSS Attacks Enabling Unauthorized Cookie Access, Section 4 shows a comprehensive study on XSS, Section 5 discusses tools, Section 6 ends with a discussion on future trends and Section 7 gives a brief conclusion.

2. CATEGORIES OF XSS ATTACKS

After reviewing several research papers, we have found that three specific categories of XSS attacks are the most mentioned- Stored XSS, Reflected XSS, and DOM-based XSS [9] [10] [11]. These three XSS attack types are well-known and widely documented since they address the most typical attack vectors along with important security consequences. These types of attacks are fairly simple to execute and they may severely impact users. Though these three types of attacks have slightly varied methods for targeting web applications, they all aim at stealing information about user accounts as the end goal. Most of the research papers we have come across primarily focus on the client-side detection of XSS attacks rather than server-side XSS. The following provides a detailed explanation and illustration of each of the categories mentioned above.

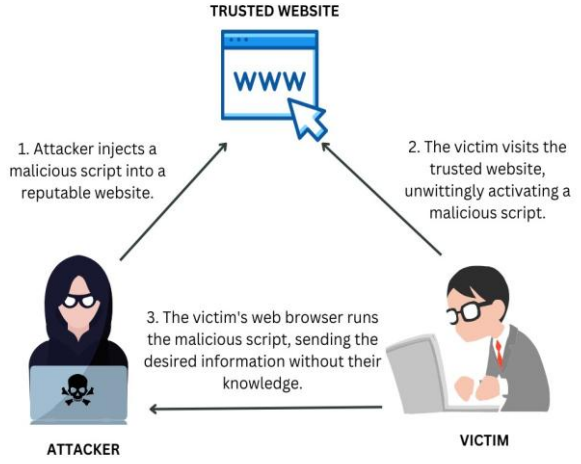

Fig. 1. XSS Workflow.

2.1 The Stored XSS (SXSS) Attack

A SXSS attack is persistent XSS and involves injecting malicious JavaScript or code into a web application through an input field like a comment box or user profile [9]. The server stores this malicious payload, which is then served to more users who visit the infected page. SXSS attacks are the most hazardous since they can affect all users of the web application, not just those interacting with a malicious link or page. These vulnerabilities can be used by attackers to steal private data from users, sabotage websites, transmit malware, or use compromised accounts for further attacks. As shown in Figure 2, the attacker injects malicious code into the application, which resides on the server side. When the user requests content, they unknowingly receive this malicious code, ultimately leading to the transmission of sensitive information to the attacker.

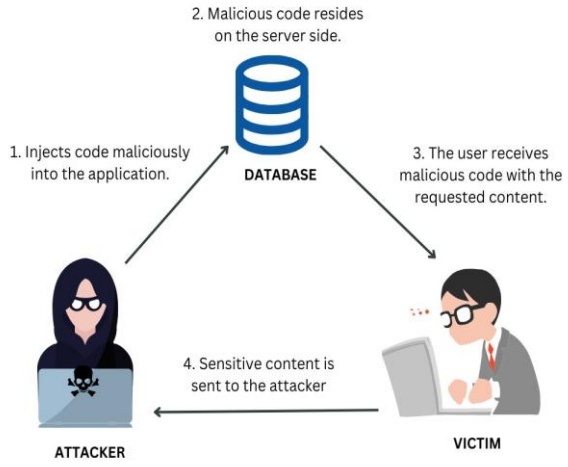

Fig. 2. SXSS Workflow.

2.2 The Reflected XSS (RXSS) Attack

A RXSS attack is a non-persistent attack that occurs when an attacker inserts malicious code into a web application's input, which is then instantly reflected by the user in the application's response [10]. This attack frequently depends on deceiving users into clicking on a malicious link or engaging with a tampered URL, which prompts their browser to run the injected script in their context. Here, only the user's request and response cycle contain the payload in a RXSS attack. There is no server-side storing of malicious scripts in this attack scenario. As depicted in Figure 3, the attacker sends a malicious URL, prompting the user to click the link and initiate its execution in the browser, which then forwards data to the attacker.

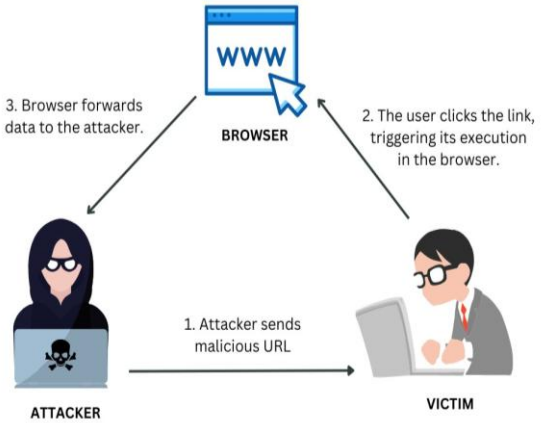

Fig. 3. RXSS Workflow.

2.3 The Document Object Model-Based (DOMB-XSS) Attack

The DOMB-XSS attack occurs on the client side of web applications and targets the document object model of a web page directly within the user's browser [11]. When a user interacts with the modified page, the injected script is run in their browser, which may result in the theft of sensitive data, session hijacking, or something else. DOMB-XSS attacks are distinct from other forms of XSS due to their independence from any communication with the server. DOMB-XSS are often regarded as non-persistent.

As illustrated in Figure 4, the attacker creates and dispatches a URL containing a malicious string. Subsequently, the user is deceived into opening the link, leading to a website request that doesn't initially respond with the malicious string. However, the user's browser executes a legitimate script, unwittingly introducing the malicious script to the page. As a result, the client-side code runs the malicious script, ultimately leading to the transmission of the user's sensitive content to the attacker.

Unauthorized access to cookies facilitated by XSS attacks [6] [12]. XSS attacks, notorious for their ability to compromise web security, often involve the theft of cookies from a user's browser. Through the injection of malicious scripts into susceptible websites, attackers can illicitly access stored cookies. Once they do this, they can take cookies that store important information like login details. This lets them pretend to be users and maybe even take control of their accounts. The exploitation of XSS vulnerabilities emphasizes the vital necessity of instituting resilient security measures to shield user data and counteract such malevolent actions.

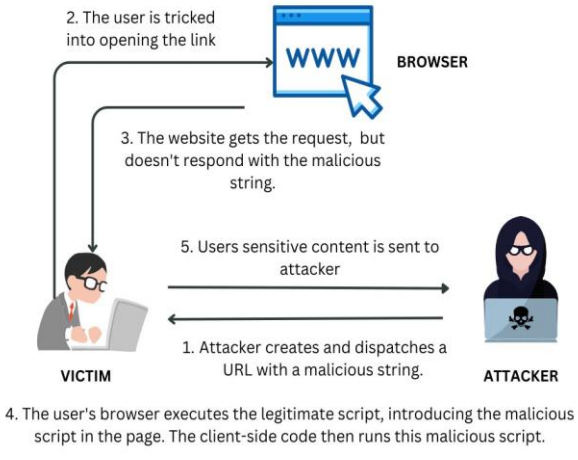

Fig. 4. DOMB-XSS Workflow.

3. COOKIES AND THEIR IMPACT IN XSS

Cookies within a web browser consist of small data pieces that websites store on a user's device. Originating from a web server, these pieces of information are transmitted to the user's browser and stored there. Cookies serve diverse purposes, ranging from tracking user preferences and sustaining login sessions to delivering a personalized browsing experience. Commonly utilized for authentication, user preference retention, and user behavior tracking for analytics, cookies also play a crucial role in tracking and analytics by collecting data on user behavior to provide insights into their interaction with a website's content. Certain cookies, often from third-party services like advertisers or social media platforms, even track users across various websites. Despite their valuable contributions to enhancing user experiences, concerns about privacy and security have prompted increased scrutiny and regulations governing how websites utilize and manage cookies. Understanding the pivotal role of cookies in web browsing is essential for users to make informed decisions regarding their privacy. Simultaneously, web developers must ensure responsible and secure data practices. In response to privacy concerns, web browsers provide controls and settings, allowing users to manage and delete cookies. Privacy modes, such as incognito or private browsing, limit the storage of cookies during a session, thereby bolstering user privacy. Implementing security measures, such as secure cookies transmitted exclusively over encrypted (HTTPS) connections and the HttpOnly flag preventing access through client-side scripts, mitigates the risk of XSS attacks. Advancements in cookie management technologies include attributes like SameSite, regulating when cookies are sent with cross-site requests. Additionally, ongoing progress in fingerprinting protections aims to bolster user privacy.

After reviewing numerous research papers and consulting various internet sources, these are the common types of cookies we have encountered- Session Cookies, Persistent Cookies, Secure Cookies, HttpOnly Cookies, First-Party Cookies, Third-Party Cookies, Analytics Cookies, Advertising Cookies, SameSite Cookies, Flash Cookies (Local Shared Objects).

3.1 Session Cookies

Session cookies, essential for web functionality, are temporary data stored on a user's device during a single browsing session. Recent statistics reveal their widespread usage, constituting approximately 65% of all cookie types employed by websites. These cookies enable smooth navigation, temporarily retaining user data and expiring upon browser closure. While privacy concerns persist, their prevalence underscores their pivotal role in improving user experience and sustaining diverse online functionalities [13].

3.2 Secure Cookies

Essential for enhanced online security, secure cookies are transmitted solely over encrypted (HTTPS) connections to protect sensitive user data. Recent statistics highlight their growing prevalence, comprising nearly 40% of cookies on

privacy-focused websites. By thwarting unauthorized access, secure cookies mitigate potential risks of data interception, contributing to bolstered cybersecurity. Their widespread adoption underscores a commitment to safeguarding user information in the evolving landscape of online threats. As cybersecurity concerns persist, the extensive use of secure cookies signifies a continual endeavor to establish a more secure and trustworthy online environment, prioritizing user protection during digital interactions.

3.3 HttpOnly Cookies

HttpOnly cookies play a crucial role in web security, designed to prevent access by client-side scripts, thereby minimizing the risk of XSS (XSS) attacks. Recent statistics underscore their importance, with over 60% of websites incorporating HttpOnly flags in their cookies to bolster security measures. By restricting script access, HttpOnly cookies strengthen defenses against potential unauthorized entry to sensitive information. Their widespread use reflects a proactive stance in addressing security vulnerabilities and safeguarding user data. As cyber threats persist, the integration of HttpOnly cookies stands as a significant measure to fortify web applications against potential exploits using scripting techniques.

3.4 First-Party Cookies

First-party cookies, crucial for personalized web experiences, are data stored by the specific website a user directly visits. Recent statistics highlight their widespread usage, making up the majority approximately 70% of all cookies employed by websites. These cookies facilitate essential functions such as user authentication, preference storage, and retaining shopping cart items. Their prevalence underscores their vital role in delivering seamless and tailored online interactions. Despite privacy considerations, first-party cookies remain widely utilized, emphasizing their crucial contribution to enhancing user experiences across diverse websites. The statistics affirm their extensive adoption, underscoring their significance in providing user-friendly and customized online browsing experiences.

3.5 Third-Party Cookies

Third-party cookies, recognized for cross-site tracking, are data files from domains different from the one a user directly visits. Recent statistics underscore their widespread usage, making up nearly 25% of all cookies employed by websites. Primarily utilized in online advertising, these cookies enable advertisers to track user behavior across multiple sites for targeted ad delivery. Despite privacy concerns, third-party cookies persist, contributing to personalized ad experiences. The statistics highlight their prevalence, indicating their significant role in shaping online advertising practices and delivering user-specific content. However, ongoing discussions and regulations aim to address privacy concerns associated with the use of third-party cookies.

3.6 Analytics Cookies

Analytics cookies, essential for website analysis, collect user data to offer insights into online behavior. Recent statistics highlight their common use, constituting a significant portion of around 20% of all cookies employed by websites. These cookies play a critical role in assisting businesses in understanding user interactions, refining content, and improving overall website performance. Despite privacy considerations, analytics cookies are widely used, showcasing their crucial role in enhancing user experiences. The statistics confirm their substantial presence, emphasizing their contribution to well-informed decision-making for optimizing websites. Amid ongoing privacy discussions, analytics cookies remain a foundational element for businesses seeking to comprehend and enhance their online presence.

3.7 Advertising Cookies

Advertising cookies, crucial for targeted marketing, gather user data to deliver personalized ads based on browsing habits. Recent statistics underscore their significant impact, making up a notable portion of around 15% of all cookies utilized by websites. These cookies empower advertisers to customize ad content, creating more personalized and relevant experiences for users. Despite privacy considerations, advertising cookies persist, enhancing the effectiveness of online advertising campaigns. The statistics highlight their common use, underscoring their pivotal role in shaping digital marketing and providing users with more personalized and engaging ad experiences. Ongoing discussions explore finding a balance between personalization and safeguarding user privacy.

3.8 SameSite Cookies

SameSite cookies play a vital role in privacy and security, featuring an attribute that controls when cookies are sent with cross-site requests. Recent statistics highlight their increasing adoption, with over 80% of websites incorporating SameSite attributes in their cookies to improve security. This attribute helps prevent cross-site request forgery (CSRF) attacks and safeguards user data from potential security risks. Despite their significant role in enhancing security, ongoing discussions focus on finding the right balance between privacy and functionality in the continually evolving landscape of web technologies. The statistics confirm their growing prevalence, showcasing a proactive approach to addressing security concerns in online interactions.

Table 1. presents a compilation of the research papers we've assessed, highlighting both their strengths and weaknesses.

Reference	Dataset	Method	Advantages	Limitations
[31]	Dmoz, ClueWeb09	Naive Bayes, SVM classifier	This experiment demonstrates an increase in PPP (Positive Predictive Power) for their features, with obfuscation-based features proving to be the most relevant.	They couldn't access the prior author's training database, limiting their ability to conduct in-depth effectiveness analyses of the employed techniques.
[27]	GITHUB	RF, SVM(L), SVM(P), LR and k-NN	This strategy enhances preparedness for evolving security challenges by identifying and blocking malicious HTTP requests and IP addresses to prevent future XSS attacks.	The research's reliance on specific algorithms, such as Random Forest (RF), may limit the method's adaptability to different machine learning models or variations.
[24]	Cannot be determined	ML with n-Gram Method	The top accuracy results were attained by SVM combined with n-gram (98%), followed by NB with n-gram (94%), and KNN with n-gram (92%).	K-Fold Cross-Validation was employed to assess machine learning on a small dataset susceptible to overfitting, preventing the memorization of noise in the training data.
[9]	Can not be determined	Cryptography	Encrypted cookies can resist XSS attacks, reducing the potential for attackers to exploit stolen cookies.	Encryption adds complexity to cookie management and users' inability to identify harmful cookies.
[16]	Can not be determined	Convolutional Neural Network	The approach adopted encompasses all dataset characters, their sequence, and order, hence more suited for ever-evolving XSS classification.	Converting XSS scripts to Unicode and removing non-ASCII characters may result in the loss of crucial contextual information and nuances from the original scripts.
[30]	Kaggle and various sources on Git Hub.	Linguistic computation & feature selection	They achieved a high accuracy of 99.87% and reduced false positives to just 0.039%.	Less flexibility can lead to small and noisy datasets.
[20]	Kaggle's dataset	SVM, KNN, Random Forest, Logistic Regression	By extracting features from both URLs and JavaScript code, a robust foundation is established for identifying potential vulnerabilities.	The approach could potentially struggle with recognizing more sophisticated new attacks.
[2]	Cannot be determined	HTML, PHP	User input from web forms is validated and sanitized protecting it from potential malicious scripts.	Regular expressions and data sanitization can incur extra processing overhead.
[18]	Cannot be determined	Intrusion detection system	Exhibits high accuracy in detecting XSS attacks.	Risk of generating false positives leading to operational overhead.
[10]	Kaggle’s dataset, GITHUB.	LR, Linear Discriminant Analysis, K-Neighbors, Decision Tree, GaussianNB, AdaBoost, Gradient	Consistent behavior between the learning curve and the verification curve suggests strong performance of the NLP-SVM model.	Low detection rate, high false-positive alerts, or high false-negative alerts.

		Boosting, RF		
[13]	Kaggle's, GITHUB	Light GBM, XGBoost, and AdaBoost	Using JavaScript developers can modify the web pages on the client's side without the need for sending a request to the server.	The efficiency of the dataset used is not effective enough, resulting in a limited payload.
[32]	Can not be determined	ASP	The validation facility blocks attacks by disallowing unencoded HTML/XHTML content processing unless explicitly permitted by the web developer.	Web developers are usually not aware that their web applications are open to script injection attacks.
[19]	Cannot be determined	AdaBoost, bagging with SVM, Extra-trees, gradient boosting, and histogram-based gradient boosting	After comparing it with multiple classifiers, it proved to have a higher accuracy rate.	Detection of new types of attack cannot be determined.
[25]	Kaggle dataset and GitHub dataset	Random Forest Algorithm, Gradient Boosting, and Decision Tree Bagging.	Improves stability, and accuracy, and reduces variance, effectively prevents overfitting. This contributes to a more reliable and robust model.	Relies on a variety of data sources, potentially leading to data heterogeneity issues and detection is based only on the client-side
[26]	Kaggle's Dataset.	Support vector technique, decision tree, Naive Bayes classifier, and logistic regression	The decision tree method produced the best results.	Difficulty dealing with noise and identified fewer malicious instances, high number of false positives, and imbalanced data.
[4]	Kaggle dataset	LSTM, CNN, AdaBoost classifier, Gradient Boosting classifier (GB), SVMs, KNN, NB, LR, RF, and Decision Tree Classifiers.	Achieve high levels of accuracy in detecting XSS attacks and can work autonomously.	Emphasizes the need to minimize false negatives. Potential trade-off in the form of increased false positives, which can disrupt valid web traffic
[3]	GitHub and Kaggle	NB, SVM, k-NN, RNN, CNN, LSTM and SE	Accuracy exceeding 99% for both detection and defense methods in real-time environments	Implementation of a hybrid system like this in a real-world production environment can be complex.
[15]	Kaggle	Adaboost, Logistic Regression, Naive Bayes, XGBoost, Decision Tree	Achieving a detection accuracy of 99.92 percent with AdaBoost	Training and using boosted models can require a significant amount of computational resources and memory
[11]	DMOZ database	ADTree, AdaBoost, SVM, and LSTM-Attention	The use of word2vec improves the model's understanding of the input data and enhances detection accuracy.	LSTM-Attention can be computationally intensive and may require substantial resources for training and inference.
[7]	Can not be determined	word2vec	While existing research compared with achieved a maximum accuracy of 98%, the proposed approach suggests	The proposed method, which combines multiple classifiers and involves preprocessing with word2vec, can be more complex and

			higher accuracy, up to 99.89%.	resource-intensive
[12]	GitHub and exploit-db	SVM, WAF	Results in enhanced detection.	Non-linear SVM, while effective, can be computationally intensive and may require more resources, potentially impacting real-time application performance.
[21]	GitHub	Stacking Ensemble	The combination of base learners, like classification algorithms and Stacking Ensembles, accurately forecasts vulnerable URLs in XSS.	The dataset used here is unbalanced and is not a benchmark dataset.
[22]	XSS-Attacks-2019 dataset	Hybrid learning approach	Secured the highest performance metrics, displaying a remarkable 99.8% achievement.	Combining models can increase computational demands, affecting system performance and response times.
[23]	Can not be determined	SVM, k-NN, Random Forest	Indicate that classifier-based methods can serve as a potent tool for identifying XSS attacks.	Risk of overfitting since two different datasets have been used.

3.9 Flash Cookies

Flash cookies, or Local Shared Objects (LSOs), store user data via Adobe Flash Player, preserving website preferences and tracking user behavior. Recent statistics reveal a gradual decrease in their prevalence, accounting for just 5% of all cookies on websites. While initially popular for persistent tracking, advancements in web technologies and increased awareness of user privacy have led to a reduced reliance on Flash cookies. The declining usage indicates a changing landscape where alternative technologies and heightened privacy concerns significantly influence how user data is stored and managed in the digital domain.

4. CURRENT STATE OF ART OF XSS

We have studied several research papers on cybersecurity, with approximately half of them focusing on the detection of XSS attacks. After going through all these papers very carefully, we have created a detailed summary of some recently published papers listed in Table 1, highlighting the most important points we've learned from these papers. These research articles collectively contribute to a deeper understanding of the evolving landscape of cyber threats, and effective defense strategies.

A series of recent research articles have delved into the intricate world of cyber security, each offering unique insights and perspectives on the ever-evolving landscape of cyber threats and defense strategies. In [16] discussed cyber-attacks and countermeasures, with a focus on categorizing attacks into hotspot and non-hotspot groups and proposing a security index for evaluating countermeasures. The importance of customizing defense strategies based on the security index and the attack stage is highlighted. In [17] delves into the growing concern of security incidents in today's complex digital landscape, pointing out the lack of guidance for organizations in learning from these incidents. The paper presents 14 research questions aimed at understanding the causes and implementing improvements.

In this series of studies spanning several years, diverse approaches to enhancing cybersecurity and identifying threats are explored. In [19] focused on the real-time detection of hybrid FDI/jamming attacks in the smart grid, incorporating Kalman filters and CUSUM-based algorithms. In [20] addressed XSS attack detection by integrating AI algorithms with cryptography to secure cookies. In [21] provides an efficient LSTM-attention-based XSS attack detection method. In [22] introduced a multi-objective reinforcement learning environment for finding XSS flaws in web applications. In [23] showcased the AdaBoost classifier's efficiency in detecting XSS payloads. In [24] employed a Convolution Neural Network (CNN) for accurate detection using various datasets. In [25] introduces a dynamic feature selection method for XSS detection, combining incremental learning and knowledge updates through a DQN-MAFS framework. These studies collectively contribute to the advancement of cybersecurity, leveraging ML techniques to enhance security, detect threats, and protect against cyberattacks in various contexts.

In [26] introduced a signature-based detection method using SNORT IDS, which also assists in identifying SQL injection attacks, offering a network packet monitoring approach. In [27] created and evaluated ensemble machine-learning approaches that outperformed single classifiers in identifying XSS attacks on web applications, achieving an overall accuracy of 0.9989. In [28] worked on manual testing of classifiers, primarily the Random Forest

Classifier, which led to 98% accuracy in distinguishing malicious and benign web content. In [29] investigated ensemble-based machine learning to enhance cybersecurity, particularly for spotting XSS attacks, using Stacking Ensembles and various classification algorithms, showing superior performance in all parameters. Then, [30] introduced an intelligent system for detecting XSS attacks using three machine learning methods, with the hybrid learning-based system outperforming all others with a 99.8% detection accuracy and precision. These studies collectively contribute to the advancement of XSS attack detection and showcase the effectiveness of various algorithms and techniques in enhancing web application security.

A variety of methodologies and algorithms of Machine Learning are explored to enhance the detection of XSS attacks. In [31] demonstrated the efficacy of SVM, k-NN, and Random Forest classifiers in identifying JavaScript XSS, achieving high accuracy and precision through feature selection. In [32] enhanced XSS detection by combining machine learning algorithms with n-gram analysis of script features, utilizing Python for mathematical libraries, and resampling for performance evaluation. In [33] used four machine learning algorithms to test for XSS detection in JavaScript, the decision tree ended up performing the best. In [34] used a combined approach for XSS detection and categorization, implementing various algorithms and security measures. In [10] introduced a new hybrid stacking ensemble technique with high accuracy, using multiple datasets and models. In [35] employed anomaly-detection methods and SVM to identify malicious JavaScript code with high detection accuracy. These studies collectively contribute to the advancement of XSS attack detection and showcase the effectiveness of various algorithms and techniques in enhancing cybersecurity.

Some researchers used hybrid approaches for the detection of XSS attacks. In [36] introduced a hybrid cyber-attack model, focusing on the combination of availability and integrity threats, which is particularly valuable for researching complex cyber-physical power systems and assessing attack tactics within resource constraints. On the other hand, [37] proposed a hybrid features model to categorize XSS attacks, achieving an impressive 99.87% accuracy with no false positives. The research selected a substantial dataset to validate the model's effectiveness, including sources like the Kaggle dataset and other datasets from GitHub. While [36] emphasizes the modeling of cyber-physical power systems and attack tactics, [37] concentrates on improving the categorization of XSS attacks, illustrating how different facets of cybersecurity research contribute to a broader understanding of threat detection and mitigation in the digital landscape.

We conducted a thorough examination to determine the number of research papers specifically addressing XSS over the past decade, spanning from 2013 to 2022. Figure 5 represents the outcomes of this inquiry with a graphical illustration for a detailed and accessible interpretation of the data.

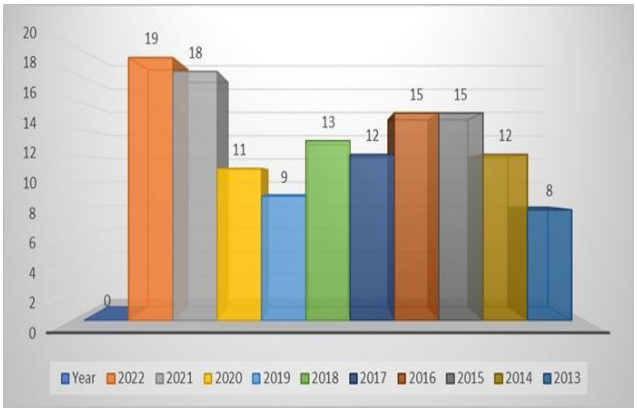

Fig. 5. Annual Publication Trends in XSS.

From the above graph on the annual publication trends of XSS research papers spanning the years 2013 to 2022, we have drawn several conclusions. Firstly, there exists a consistent and enduring interest in XSS throughout the entire decade, as evidenced by the sustained publication of research papers. The observed fluctuations in the number of publications each year indicate a dynamic and evolving landscape of research activity in the field. Noteworthy peaks in 2015, 2016, 2021, and 2022 suggest periods of intensified focus, potentially corresponding to significant advancements, increased awareness, or heightened security concerns. Conversely, the relatively stable output during the years 2017 to 2020 implies a sustained but comparatively steady level of research activity. The recent surge in publications in 2021 and 2022 further underscores a renewed or heightened interest, potentially indicative of emerging developments or breakthroughs in XSS research. In summary, the data paints a nuanced picture of the evolving dynamics and varying levels of engagement in XSS research over the specified ten-year period.

5. ANALYSING TOOLS OF XSS DETECTION

Various tools like Qualys Guard, Escaner Web, and Detect Intrusions gauge URL safety [38], allowing or disallowing their execution. However, they're not entirely functional, requiring manual scans of necessary URLs by the user. Acunetix WVS, Rational AppScanEnterprise, and ZAP [39] are black box web scanners. They enable custom test bench development to compare capabilities but perform relatively poorly in detecting only stored XSS. Ghost.py [40] is a browser without headers that interprets JavaScript and simulates browser behavior to uncover hidden injection points. Although accessible for secondary development, it relies on pre-established vulnerabilities for attack. Kameleon Fuzz [41] combines evolutionary fuzzing with inference models to generate test cases for XSS type 1, but

it lacks real-world application experience. Web Input Vector Extractor Teaser (WIVET) [42] serves as a distributed scanning tool providing a database of attack vectors. However, it only analyzed the top 1000 Alexa-ranked websites. Dom XSS Micro focusing on benchmarking, aims to extract representative vulnerabilities for templates but remains incomplete [43, 62, 63, 64, 65, 66]. The GIT Repository, a static analysis approach to detect SQL injection and cross site scripting vulnerabilities in web applications, using text mining, detects XSS vulnerabilities in code files [44, 56]. Its programming language version is independent but insufficient for developing machine learning models. XSSDM, towards detection and mitigation of XSS vulnerabilities in web applications, analyzes patterns via C hash prototype to minimize false positive and negative results, limited to PHP-based web applications [45]. RIPS & PIXY, perform detailed defensive programming analysis, aiming to precisely detect XSS vulnerabilities [46, 57, 58]. Yet, only one empirical evaluation has been presented. Fiddler [47], based on Kullback-Leibler Divergence, detects known XSS attack signatures with low false positives, tested on only three vulnerable PHP applications.

Pearl-based IDS [48] captures and hashes executable content, suitable for user-controlled sites but tested only on the authors created web pages. XSSERC [49] employs rule-based filters to estimate patterns in intercepted requests but only tests performance on the top 50 global websites. Snort [50] utilizes regular expression patterns for XSS attack investigation but faces challenges with false positives, resource usage, and lacks preventive capabilities. WebKit XSS Auditor [51] focuses on PHP code vulnerabilities, presenting examples of poorly written PHP code as case studies. ETSS Detector [52] automatically identifies XSS vulnerabilities but is limited to detecting persistent and non-persistent XSS only. GIT Repository [45] utilizes feature extraction algorithms for machine learning models but lacks vulnerability prediction characteristics. PURITY [53] detects SQL and XSS injection threats based on planned execution, with minimal manual intervention, but only one case study has been evaluated. Grease Monkey filters [43] text and analyzes syntax, yet faces compatibility issues with DOM manipulation operations, potentially omitting vulnerable scripts. .Net [54] attacks systems via web requests, aiming to minimize false positives and bolster input disinfection, but it remains a proposed study. Static Analysis Tools [55, 59, 60, 61] measure the performance of diverse configurations, but there's a potential increase in false positives.

The extensive range of tools utilized by researchers in their investigations on XSS reveals a broad spectrum of approaches adopted to combat this security threat. These tools, encompassing solutions like Qualys Guard, Acunetix WVS, Ghost.py, Snort, and more, indicate the availability and diverse functionalities designed to tackle XSS vulnerabilities. However, each tool demonstrates its own set of limitations, from partial functionality requiring manual scans to limitations in detecting specific types of XSS attacks. Some tools, such as Acunetix WVS or RIPS, specialize in precise XSS vulnerability detection, while others like Web Input Vector Extractor Teaser (WIVET) focus on analyzing attack vectors within websites. Despite their functionalities, many tools undergo testing on limited website sets or controlled environments, potentially affecting their real-world applicability and effectiveness. Notably, some tools emphasize prevention by reinforcing web applications against potential attacks, while others concentrate on detecting existing vulnerabilities. Challenges like false positives, detection limitations, and compatibility issues emerge, raising concerns about the tools' overall performance and their practical application in diverse web environments. This collective usage of tools underscores the intricate nature of addressing XSS vulnerabilities, advocating for a multifaceted approach that encompasses detection, prevention, and continual validation to fortify web security effectively.

6. FUTURE TRENDS AND DISCUSSION

After our thorough study on XSS we believe presently, XSS research is exploring new ways attackers use to breach security, improving methods to detect such attacks, and finding ways to secure modern web technologies and there's a strong emphasis on the development and enhancement of tools for detecting, preventing, and mitigating XSS vulnerabilities. Researchers are increasingly using machine learning and AI to better spot and stop these threats. Looking forward, XSS research might focus on AI-based defenses that actively prevent attacks by analyzing attack patterns. The trajectory of XSS tool development might pivot towards more sophisticated AI powered solutions capable of intelligently identifying and neutralizing XSS threats in real-time. Additionally, there could be an expansion in the integration of security tooling directly into development environments, enabling developers to build and test applications with XSS prevention measures seamlessly integrated. The future might also witness the emergence of more comprehensive tool suites that encompass both detection and remediation functionalities, offering a holistic approach to XSS mitigation. As technology advances, new frontiers like the Internet of Things (IoT) and 5G networks are shaping our digital world. Future XSS research aims to understand how these technologies might introduce fresh security challenges. For instance, IoT devices constantly exchange data, potentially creating new avenues for XSS attacks. Additionally, the faster and more connected 5G networks could amplify the impact of XSS vulnerabilities. This exploration widens the horizon of XSS research, ensuring we fortify our defenses against emerging threats in this evolving digital landscape. XSS research is evolving to detect and prevent attacks while enhancing tools using AI

and machine learning. This prepares us for evolving digital challenges in XSS security.

7. CONCLUSION

Due to weak security execution, XSS vulnerabilities still exist despite being an older style of web-based application attack. This paper provides a comprehensive survey of XSS attacks and their related defense strategies discovered in research papers published from 2010 to 2023. The study reviews 70 papers and identifies a prevalent focus on client and server-side XSS solutions, emphasizing the effectiveness of hybrid approaches. Moreover, 20 research papers have been studied directly focusing on analyzing the tools used by the researchers. Recent trends show the integration of machine learning techniques in XSS prevention, which have proven effective in detecting unknown attacks. Also, we have determined that hybrid and stacking approaches consistently yield significantly superior results compared to using homogeneous classifiers alone. The paper underscores the importance of developing robust solutions to combat evolving XSS variants. The survey results aim to enhance the understanding of XSS protection measures and guide the development of more holistic security solutions. The diversity of tools used for XSS research highlights varied approaches, but limitations like incomplete coverage and performance challenges persist. To fortify web security effectively, a holistic strategy combining prevention, detection, and thorough testing across diverse web environments is imperative. Overall, this study contributes significantly to the development of defensive mechanisms crucial for safeguarding rapidly expanding web applications. It highlights the ongoing threat of XSS and stresses the need for user awareness, secure software development, and consistent safeguard maintenance to prevent critical service and data breaches.

REFERENCES

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

[2] Li Y, Liu Q. 2021. A comprehensive review study of cyber-attacks and cyber security; emerging trends and recent developments. Energy Reports 7, 8176–8186

[3] Tariq I, Sindhu MA, Abbasi RA, Khattak AS, Maqbool O, Siddiqui GF. 2021. Resolving XSS attacks through genetic algorithm and reinforcement learning. Expert Systems with Applications 168, 114386

[4] Marashdih AW, Zaaba ZF. 2017. Cross site scripting: Removing approaches in web application. Procedia Computer Science 124, 4th Information Systems International Conference, ISICO, Bali, Indonesia 647–655

[5] Wu A, Feng Z, Li X, Xiao J. Ztweb: 2023. Cross site scripting detection based on zero trust. Computers & Security 134, 103434 [6] Rodr´ıguez GE, Torres JG, Flores P, Benavides DE. 2020. XSS (xss) attacks and mitigation: A survey. Computer Networks 166, 106960

[6] Malviya VK, Rai S, Gupta A. 2021. Development of web browser prototype with embedded classification capability for mitigating XSS attacks. Applied Soft Computing 102, 106873

[7] Mokbal FMM, Dan W, Xiaoxi W, Wenbin Z, Lihua F. 2021. Xgbxss: An extreme gradient boosting detection framework for XSS attacks based on hybrid feature selection approach and parameters optimization. Journal of Information Security and Applications 58, 102813 [9] Taha TA, Karabatak M. 2018. A proposed approach for preventing XSS. In: 2018 6th International Symposium on Digital Forensic and Security (ISDFS), pp. 1–4

[8] Krishnan M, Lim Y, Perumal S, Palanisamy G. 2022. Detection and defending the xss attack using novel hybrid stacking ensemble learning-based dnn approach. Digital Communications and Networks

[9] Harish Kumar J, J Godwin Ponsam J. 2023. Cross site scripting (xss) vulnerability detection using machine learning and statistical analysis. In: 2023 International Conference on Computer Communication and Informatics (ICCCI), pp. 1–9

[10] Hydara I, Sultan ABM, Zulzalil H, Admodisastro N. 2015. Current state of research on XSS (xss) – a systematic literature review. Information and Software Technology 58, 170–186

[11] Moataz A. Ahmed, Fakhreldin Ali (2016) Multiple-path testing for XSS using genetic algorithms. Journal of Systems Architecture 64, 50–62

[12] Wang R, Xu G, Zeng X, Li X, Feng Z. Tt-. 2018. xss: A novel taint tracking based dynamic detection framework for dom XSS. Journal of Parallel and Distributed Computing 118, 100–106

[13] Marashdih AW, Zaaba ZF, Suwais K, Mohd 2018. Web application security: An investigation on static analysis with other algorithms to detect cross site scripting. Procedia Computer Science 161, 1173–1181 The Fifth Information Systems International Conference, 23-24 July 2019, Surabaya, Indonesia

[14] Wu Y, Ru Y, Lin Z, Liu C, Xue T, Zhao X., Chen J. 2023. Research on cyber-attacks and defensive measures of power communication network. IEEE Internet of Things Journal 10(9), 7613–7635

[15] Patterson CM, Nurse JRC, Franqueira VNL. 2023. Learning from cyber security incidents: A systematic review and future research agenda. Computers & Security 132, 103309

[16] Akaishi S, Uda R. 2019. Classification of xss attacks by machine learning with frequency of appearance and co-occurrence. In: 2019 53rd Annual Conference on Information Sciences and Systems (CISS), pp. 1–6

[17] Kurt MN, Yılmaz Y, Wang X. 2019. Real-time detection of hybrid and stealthy cyber-attacks in smart grid. IEEE Transactions on Information Forensics and Security 14(2), 498–513

[18] Mishra P, Gupta C. 2020. Cookies in a XSS: Type, utilization, detection, protection and remediation. In: 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), pp. 1056–1059

[19] Lei L, Chen M, He C, Li D. 2020. XSS detection technology based on LSTM attention. In: 2020 5th International Conference on Control, Robotics and Cybernetics (CRC), pp. 175–180

[20] Caturano F, Perrone G, Romano SP. 2021. Discovering reflected XSS vulnerabilities using a multi objective reinforcement learning environment. Computers & Security 103, 102204
[21] Roy P, Kumar R, Rani P, Joy TS.: 2022. Xss: XSS attack detection by machine learning classifiers. In: 2022 11th International Conference on System Modeling & Advancement in Research Trends (SMART), pp. 1535–1539
[22] Kumar J, Santhanavijayan A, Rajendran B. 2022. Cross site scripting attacks classification using convolutional neural network. In: 2022 International Conference on Computer Communication and Informatics (ICCCI), pp. 1–6
[23] Kareem Thajeel I, Samsudin K, Jahari Hashim S, Hashim F. 2023. Dynamic feature selection model for adaptive cross site scripting attack detection using developed multi-agent deep q learning model. Journal of King Saud University - Computer and Information Sciences 35(6), 101490
[24] Gupta K, Ranjan Singh R, Dixit M. 2017. Cross site scripting (xss) attack detection using intrustion detection system. In: 2017 International Conference on Intelligent Computing and Control Systems (ICICCS), pp. 199–203
[25] Nagarjun P, Ahamad SS. 2020. Ensemble methods to detect xss attacks. International Journal of Advanced Computer Science and Applications 11(5)
[26] Banerjee R, Baksi A, Singh N, Bishnu SK. 2020. Detection of xss in web applications using machine learning classifiers. In: 2020 4th International Conference on Electronics, Materials Engineering & Nano-Technology (IEMENTech), pp. 1–5
[27] Perumal S, Sujatha PK. 2021. Stacking ensemble-based xss attack detection strategy using classification algorithms. In: 2021 6th International Conference on Communication and Electronics Systems (ICCES), pp. 897–901
[28] Abu Al-Haija Q. 2023. Cost-effective detection system of XSS attacks using hybrid learning approach. Results in Engineering 19, 101266
[29] Mereani FA, Howe JM. 2018. Detecting XSS attacks using machine learning. In: Hassanien, A.E., Tolba, M.F., Elhoseny, M., Mostafa, M. (eds.) The International Conference on Advanced Machine Learning Technologies and Applications (AMLTA2018), pp. 200–210. Springer, Cham
[30] Habibi G, Surantha N. 2018. Xss attack detection with machine learning and n-gram methods. In: 2020 International Conference on Information Management and Technology (ICIMTech), pp. 516–520
[31] Kascheev S, Olenchikova T. 2020. The detecting XSS (xss) using machine learning methods. In: 2020 Global Smart Industry Conference (GloSIC), pp. 265–270
[32] Chen HC, Nshimiyimana A, Damarjati C, Chang PH. 2021. Detection and prevention of XSS attack with combined approaches. In: 2021 International Conference on Electronics, Information, and Communication (ICEIC), pp. 1–4
[33] Alazab, A., Khraisat, A., Alazab, M., Singh, S.: 2022. Detection of obfuscated malicious javascript code. Future Internet 14(8)
[34] Tu, H., Xia, Y., Tse, C.K., Chen, X.: 2020. A hybrid cyber-attack model for cyber-physical power systems. IEEE Access 8, 114876–114883
[35] Prasetio K, Arief MR. 2022. XSS attack detection using machine learning with hybrid features. JURNAL INFOTEL
[36] Mehta TS, Jamwal S. 2015. Model to prevent websites from xss vulnerabilities effectiveness of black-box web application scanners in detection of stored sql injection and stored xss vulnerabilities. In: 2015 10th International Conference for Internet Technology and Secured Transactions (ICITST), pp. 186–191
[37] Liu Y, Zhao W, Wang D, Fu L. A 2015. xss vulnerability detection approach based on simulating browser behavior. In: 2015 2nd International Conference on Information Science and Security (ICISS), pp. 1–4
[38] Duchene F, Groz R, Rawat S, Richier JL. 2012. Xss vulnerability detection using model inference assisted evolutionary fuzzing. In: 2012 IEEE Fifth International Conference on Software Testing, Verification and Validation, pp. 815–817
[39] Nguyen TK, Hwang SO. 2016. Large-scale detection of dom-based xss based on publisher and subscriber model. In: 2016 International Conference on Computational Science and Computational Intelligence (CSCI), pp. 975–980
[40] Pan J, Mao X. 2017. Detecting dom-sourced XSS in browser extensions. In: 2017 IEEE International Conference on Software Maintenance and Evolution (ICSME), pp. 24–34
[41] Gupta MK, Govil MC, Singh G. 2014. Static analysis approaches to detect sql injection and cross site scripting vulnerabilities in web applications: A survey. In: International Conference on Recent Advances and Innovations in Engineering (ICRAIE-2014), pp. 1–5
[42] Gupta MK, Govil MC, Singh G. 2017. Predicting XSS (xss) security vulnerabilities in web applications. In: 2015 12th International Joint Conference on Computer Science and Software Engineering (JCSSE), pp. 162–167
[43] Gupta MK, Govil MC, Singh G. 2014. A context-sensitive approach for precise detection of XSS vulnerabilities. In: 2014 10th International Conference on Innovations in Information Technology (IIT), pp. 7–12
[44] Shahriar H, North S, Chen WC, Mawangi E. 2013. Design and development of anti-xss proxy. In: 8th International Conference for Internet Technology and Secured Transactions (ICITST-2013), pp. 484–489
[45] Frenz CM, Yoon JP. 2012. Xssmon: A perl based ids for the detection of potential xss attacks. In: 2012 IEEE Long Island Systems, Applications and Technology Conference (LISAT), pp. 1–4
[46] Bozic J, Wotawa F. 2013. Xss pattern for attack modeling in testing. In: 2013 8th International Workshop on Automation of Software Test (AST), pp. 71–74
[47] Zalbina MR, Septian TW, Stiawan D, Idris MY, Heryanto A, Budiarto R. 2017. Payload recognition and detection of cross site scripting attack. In: 2017 2nd International Conference on Anti-Cyber Crimes (ICACC), pp. 172–176
[48] Stasinopoulos A, Ntantogian C, Xenakis C. 2017. Bypassing xss auditor: Taking advantage of badly written php code. In: 2014 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), pp. 000290–000295
[49] Rocha TS, Souto E. 2014. Etssdetector: A tool to automatically detect XSS vulnerabilities. In: 2014 IEEE 13th International Symposium on Network Computing and Applications, pp. 306–309
[50] Bozic J, Wotawa F. 2015. Purity: A planning-based security testing tool. In: 2015 IEEE International Conference on

Software Quality, Reliability and Security Companion, pp. 46–55

[51] Sonewar PA, Mhetre NA. 2015. A novel approach for detection of sql injection and cross site scripting attacks. In: 2015 International Conference on Pervasive Computing (ICPC), pp. 1–4

[52] Algaith A, Nunes P, Jose F, Gashi I, Vieira M. 2018. Finding sql injection and cross site scripting vulnerabilities with diverse static analysis tools. In: 2018 14th European Dependable Computing Conference (EDCC), pp. 57–64

[53] Rawoteea Y. and Bekaroo G. 2024. "RXSS Protect: A Browser Extension for Detection of Reflected XSS Attacks in Real-Time Using Machine Learning," 2024 International Conference on Next Generation Computing Applications (NextComp), Mauritius, pp. 1-7,

[54] Alanda A., Satria D. and Mooduto H. A. 2024. XSS (XSS) Vulnerabilities in Modern Web Applications, 11th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), Yogyakarta, Indonesia, pp. 270-276,

[55] Sa'adah K., Hanaputra R. R., Girinoto and Cahyono S., 2024. A Deep Learning Approach to Detect XSS Attack as a Web Application Firewall, International Conference on Information Technology and Computing (ICITCOM), Yogyakarta, Indonesia, pp. 93-98,

[56] Hakim N. A. N, Suryani V. and Irsan M. 2024. Detection of XSS Attacks on Web Applications Using the LSTM Method, 12th International Conference on Information and Communication Technology (ICoICT), Bandung, Indonesia, 2024, pp. 432-437,

[57] Nad T. P. and Kumari S. V, 2024. An Analysis of Cross-Site-Scripting Attack with Accuracy using Support Vector Machine and Convolutional Neural Network, International Conference on Trends in Quantum Computing and Emerging Business Technologies, Pune, India, pp. 1-5

[58] Bacha U. N., Lu S., Rehman U. A, Idrees M, Ghadi Y. Y, Alahmadi J. T., 2024. Deploying Hybrid Ensemble Machine Learning Techniques for Effective XSS (XSS) Attack Detection, Computers, Materials and Continua, Volume 81, Issue 1, Pages 707-748

[59] Tariq I., Sindhu A. M., Abbasi A. R., Khattak S. A, Maqbool O, Siddiqui F. G, 2021. Resolving XSS attacks through genetic algorithm and reinforcement learning, Expert Systems with Applications, Volume 168

[60] Younas F, Raza A., Thalji N, Abualigah L, Zitar A. R, Jia H. 2024. An efficient artificial intelligence approach for early detection of XSS attacks, Decision Analytics Journal, Volume 11

[61] Thajeel K. I, Samsudin K., Hashim J. S, Hashim F, 2023. Dynamic feature selection model for adaptive cross site scripting attack detection using developed multi-agent deep Q learning model, Journal of King Saud University - Computer and Information Sciences, Volume 35, Issue 6

[62] Verma1 A., Singh A, Bihari A., Tripathi S. 2023. Identification of Hate Speech on Social Media using LSTM, GMSARN International Journal 17 PP: 468-474

[63] Kumar V., Gupta K. S., Hussain A., Sharma A. 2025.A Systematic Approach to Prevent Threats Using IDS in IoT Based Devices, GMSARN International Journal 19 PP: 107-112