A Detailed Survey on Recent XSS Web-Attacks Machine Learning Detection Techniques

Conference Paper · October 2021

DOI: 10.1109/GCAT52182.2021.9587569

2 authors, including:

Jasleen Kaur
Chandigarh University
5 PUBLICATIONS **26** CITATIONS

SEE PROFILE

A Detailed Survey on Recent XSS Web-Attacks Machine Learning Detection Techniques

Jasleen Kaur, PhD Scholar
Department of Computer Science and Engineering
Chandigarh University
Punjab, India
jasleenkaur2136@gmail.com

Dr. Urvashi Garg, Associate Professor
Department of Computer Science and Engineering
Chandigarh University
Punjab, India
urvashigarg.ibmcse@cumail.in

***Abstract*—XSS attacks have become more prevalent in last few decades and thus more challenging to detect their existence. XSS attacks are broadly classified into two categories: server-based XSS attack and client-based XSS attack. Although a lot of research has already been done in this area, still the methods lack in precision and accuracy as per the literature survey. There are ample of methodologies being applied in the detection of XSS attacks using supervised learning, unsupervised learning, reinforcement learning, deep learning and metaheuristic algorithms. We present a survey of the recent approaches being applied by the numerous researchers in their proposed models. Following indexed journals were used for research papers' collection in order to carry out a survey: Elsevier, Springer, IEEE explore, Hindawi, google scholar, and Web of Science. Moreover, in this paper, we introduce a classification chart of several machine learning algorithms that can be applied to the web-attack detection model.**

Keywords—XSS attacks, machine learning, reinforcement, tools, classifiers, k-fold cross validation, feature engineering.

I. INTRODUCTION

XSS attack is a subset of injection attack wherein an attacker injects a malicious code (also known as malicious payload) into the contents of a legitimate and trusted websites in order to gain access control of the viewer's system. XSS attack can be executed on any vulnerable website be it the one written in HTML code, JavaScript, VBScript, PHP, etc. The following steps explain the general scenario of an attack:

Step 1: Attacker finds a vulnerable website which allows the injection of untrusted malicious code into its webpage. For example, inserting false advertisement on the web page, displaying false content on the website.

Step 2: Attacker inserts malicious client-side JavaScript/ActiveX/VBScript/HTML code on the web application. This code is either sent to the victim's web browser or the web server depending upon the type of XSS attack.

Step 3: User click on the malicious link either while visiting the website or accessing service from web server.

Step 4: Attacker has access to private credentials or details of the victim through vulnerable website by bypassing the SOP (Same Origin Policy).

Nature of the attack depends upon the type of XSS attacks. XSS attacks can be classified into the following categories:

A. Persistent/ Stored XSS

Persistent XSS is also known as Stored XSS. In this attack. The malicious script is added directly on the website (especially forms, blogs or comment sections), therefore, it is also known as direct/ second-order/ type-1/ stored XSS attack as the script gets stored on the web server. So, whenever user visits that website, the malicious code gets executed, and hence, it is said to be more harmful than other two types as it does not even require the user to click on the link and can penetrate easily to multiple users at one time. Therefore, it is challenging to identify this type of attack. The following figure fig.1 represents the sample scenario of stored XSS attack.

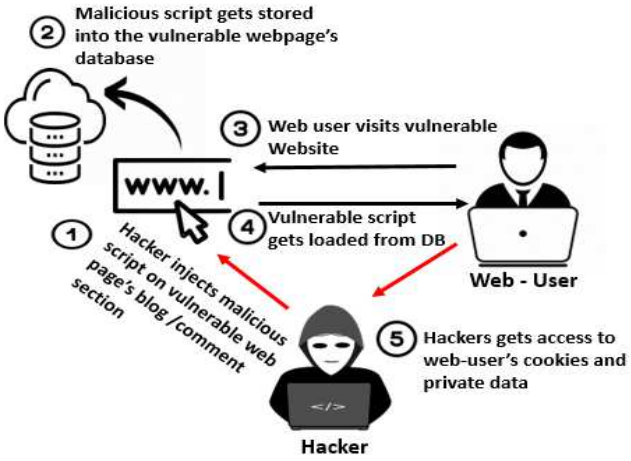

Figure 1. Sample Stored XSS attack scenario

B. Reflected/ Non-Persistent XSS

Non-Persistent XSS attack is also referred as type-II or reflected XSS attack where reflected means that the results of malicious query is visible to the attacker. The attacker crafts the malicious link in such a way that it appears to be from a trusted source. When the victim clicks on the malicious link, web server sends a response including the malicious script to the user. The following figure fig 2. represent the sample scenario of reflected XSS attack.

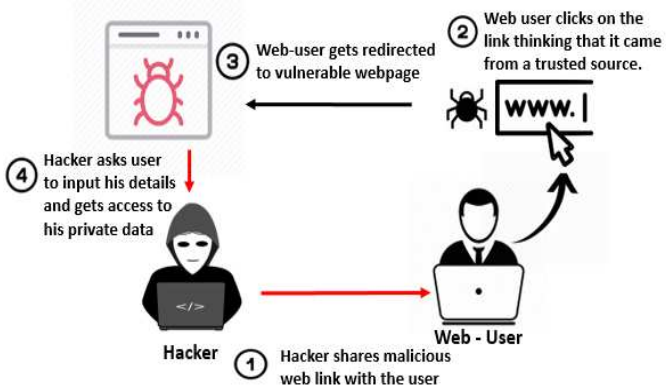

Figure 2. Sample Reflected XSS attack scenario

978-1-7281-8402-9/21/$31.00 ©2021 IEEE

C. DOM- based XSS

Another significant attack vector is the DOM-based XSS attack which has affected the world's most prominent organizations such as Google, Yahoo, and Amazon. Although its effect is moderate, it is still a serious concern for the web applications. DOM- based/ type-0 XSS attack is a client-side vulnerability attack where the attacker modifies the content of a static or dynamic DOM page at the client side by using document properties such as document.write, document.url or document.referrer. Consequently, the client-side code executes in an undesired manner and web-user will be completely unaware of the attack as there will be no alteration in the contents of web-page being displayed.

DOM means Document Object Model. It is an API which is used to access the contents or properties of a web page also known as document. In other terms, DOM is an object model for every HTML or XML based webpage. So, when DOM-based XSS attack is executed, the JavaScript code is embedded in the client-side program, thereby, modifying the contents of DOM and can also change the values of objects' properties, while the user visits the page without clicking malicious link. Since the attack vector is placed in a response page and malicious code executes on victim's computer, server-side detection algorithm would fail to detect this type of attack. The following figure fig 3. represent sample scenario of DOM based XSS attacks.

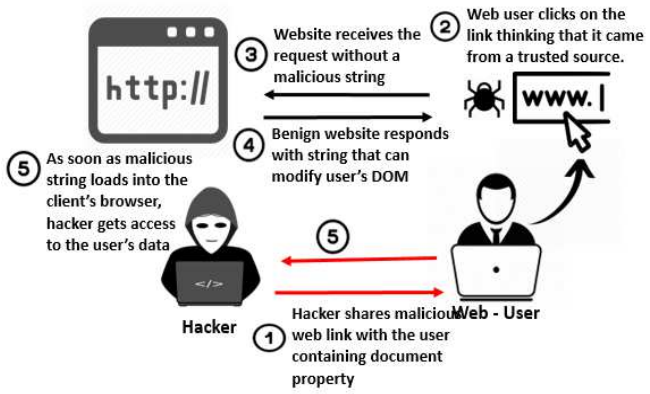

Figure 3. Sample DOM- XSS attack scenario

D. Mutation- based XSS

This is another class of XSS attacks discovered recently by PhD scholar Dr. Mario in 2018 which is the result of errors in inner HTML code and its properties which is used by the all modern browser to display the HTML content to the user before being filtered by the server. Dr. Mario in [2] introduced six new sub-classes of mXSS attacks which were not known before their study. Since mXSS is recursive in nature and the iteration is highly unpredictable, it is named as mutated XSS or mutation based XSS. The innerHTML allows the automatic modification of the HTML content and thus bypassing the DOM.

The problem arises when the content gets inserted into the browser's DOM where it mutates the stuff. Using the element.innerHTML property, when the content gets inserted into the browser's DOM, the browser mutates the content to make sure the content is error-free or there is no illegal markup. The major challenge with this type of XSS attack is that it easily bypasses the server-side defense mechanism and client-side filters. The following figure fig 4. represent sample scenario of mutation based XSS attacks.

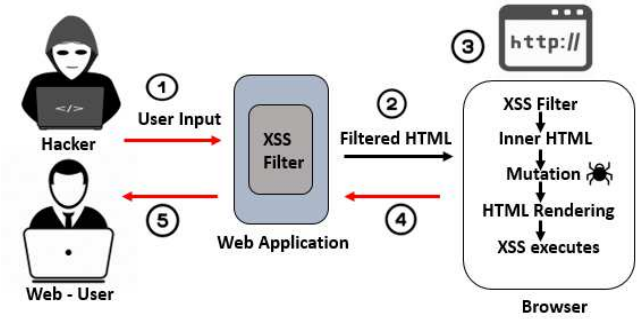

Figure 4. Sample DOM- XSS attack scenario [2]

II. DEFENESE MECHANISMS AGAINST XSS ATTACKS

Several defense mechanisms have been proposed till date for defending against XSS attacks. Some of them are mentioned below:

A. Traditional Methods

1) *Input filtering*
2) *Output escaping*

Since XSS attacks are implemented by injecting a malicious script into a vulnerable webpage, the only way to detect the illegal script from the source code is to either filter the input and escaping the output [15]. Input filtering or input validation is done primarily on the basis of white-list values and black-list values. While the first ensures that the inputted data is completely safe from any undesirable illegal input, the output escaping ensures that the application receives secure data by replacing the escape characters with other characters such as # can be replaced with &#35, < with &it, > with &gt.

B. Source Code Analysis

1) *Static Code Analysis*
2) *Dynamic Code Analysis*
3) *Hybrid Code Analysis*

While testing the web application as a security check, programmers utilize either static analysis, dynamic analysis or hybrid analysis of the code. Static analysis do not require the program to be executed and can be performed by simply inspecting each line of code for any unknown behavior, while dynamic analysis is performed by analyzing the program using the real-world input values during the program execution. Hybrid analysis involves combining the methods of static and dynamic analyses. However, these testing techniques are time-consuming and are not effective at finding new bugs in the system that may modify the content of the web page.

C. Machine Learning Defense Mechanisms

1) *Supervised Learning*
2) *Unsupervised Learning*
3) *Semi-supervised Learning*
4) *Reinforcement Learning*
5) *Deep Learning*

Although vast research has been done in detection and prevention of web attacks covering client-side protection mechanisms safeguarding the user's environment, server-side protection, and network-layer defense mechanisms, there is still lack of focus given to web-attacks detection using artificial intelligence techniques. "Noxes" was the first client-side solution proposed in 2009 by E.Kirda, WAF is a network-based defense mechanism, and modsecurity is a server-side defense. With the advancements in internet and increasing usage of websites, attackers have become more active from the last decade. XSS attack vectors have become heterogeneous in nature and are passed as hidden values in URLs, images, and script documents.

The only effective method to mitigate these attacks is to implement machine learning algorithms in the attack detection model for the best results [15]. However, the usage of artificial techniques and their sub-classes for XSS attacks detection is still at ground level and needs more attention. Machine learning is a sub-category of artificial intelligence where the algorithms learn to improve themselves automatically with the training dataset. This survey paper focuses on XSS attacks detection or prevention using machine learning approaches that had been proposed from the last 4 years (2018-2021).

III. RELATED WORK

The most prominent websites such as Google, Twitter, Facebook, YouTube, and eBay had also been affected by XSS attacks. To explain, CheckPoint Research [1] found that one of TikTok's subdomains - ads.tiktok.com - was vulnerable to XSS attacks, which lets an attacker input malicious scripts. Discussing about a vulnerability found in Facebook in the year 2020, it was a DOM-based XSS vulnerability found on the Facebook Login button. After doing the complete survey and reading 17 research papers and articles, we have presented a classification chart in this paper and captioned it as figure 5. This classification chart would help the researchers to identify the AI model that they can implement for the web-security problem resolution.

C. Stefano (2020) [3] used Mitch: a black box vulnerability detection method to detect CSRF attack which is also a client-side attack. Unlike white box detection techniques, black box method requires less time, no prior knowledge of the underlying software, and operates at the HTTP level handling both HTTP requests and responses.

The researcher states that it is possible to reinforce re-authentication or one-time password to prevent any malicious requests going unnoticed. Since cookies are sent automatically in requests to the domain that they are set for, it possess a high vulnerability risk of CSRF attack when cookies are being shared with third party website. (Cross site means from one site to another). Enabling SameSite cookie attribute, which was first introduced by Google Chrome browser, can also be helpful in preventing CSRF attack and is highly recommended for new websites. This is so because cookies will be filtered before being sent automatically.

Explaining more about SameSite cookie, it has three values: strict, lax, and none where the first type will not allow to pass through another party and requests will be blocked. The second value (lax) will allow the GET requests and block the unsecure POST requests, while the last value (none) allows all types of requests and it has to be made through secure HTTPS connection. For authentication purposes, strict or lax must be used, while for advertisement purpose, none could be used with secure tag.

IV. XSS DETECTION TECHNIQUES

You are now ready to style your paper; use the scroll down window on the left of the MS Word Formatting toolbar.

A. Exploring Dark Web Using Machine Learning (2019)

In [4], the researchers proposed a method to explore the dark web by extracting critical posts from the forums available online using machine learning and doc2vec. The doc2vec is a natural language processing algorithm which is used to extract the information / labels from the text document in order to apply it on the machine learning model. The posts related to malwares were set as the correct data and others were set as the incorrect data. The researchers used Sixgill platform in order to collect the data for their experiment and grid search algorithm to optimize the parameters.

B. Deep XSS: Deep Learning Model to Detect XSS (2018)

Yong Fang et al. (2018) [5] proposed a deep learning method to detect XSS where first the original data is restored using a decoder and then the features of XSS payloads are extracted using the word2vec. After that, the data is further fed into the LSTM neural network model. Finally, the performance of the proposed method is evaluated using 10-fold cross validation technique and is compared with ADTree and AdaBoost algorithm.

C. RLXSS: Model Based on Reinforcement Learning (2019)

Yong Fang et al. (2019) [6] proposed RLXSS: an XSS attack detection model based on reinforcement learning where this model utilizes two sub models namely, adversarial model and retraining model. This method used DDQN (dueling deep Q networks), a reward function, escape technology and XSS detection software such as SafeDog and XSSChop. The adversarial samples retrieved from adversarial model were fed into the retraining model for the optimization purpose.

D. XSS Detection Based on Supervised Machine Learning (2018)

Kaur Gurpreet et al. (2018) [7] proposed a supervised machine learning approach which detects the malicious link before it gets executed on the victim's system. Their approach uses Linear Support Vector Machine classifier for the detection of blind XSS attacks and to determine the difference between the main features of blind XSS and stored XSS attacks. While implementing features extraction phase, JavaScript events, that are utilised by the attackers to inject malicious payload, were executed. Linear separable dataset was used for experimentation purpose. Blind XSS attack has been simulated on Mutillidae which is a free vulnerable website. The detection accuracy is 95.4% with recall value 0.951 and the false positive is 0.111.

E. Preventing XSS attack using Hybrid Approach (2021)

Tariq I. et al. (2021) [8] proposed a mechanism to prevent XSS in web applications by using a hybrid methodology. They combined metaheuristic algorithm (Genetic Algorithm) with a machine learning approach and claims that this method has been used for the first time. Apart from using GA with

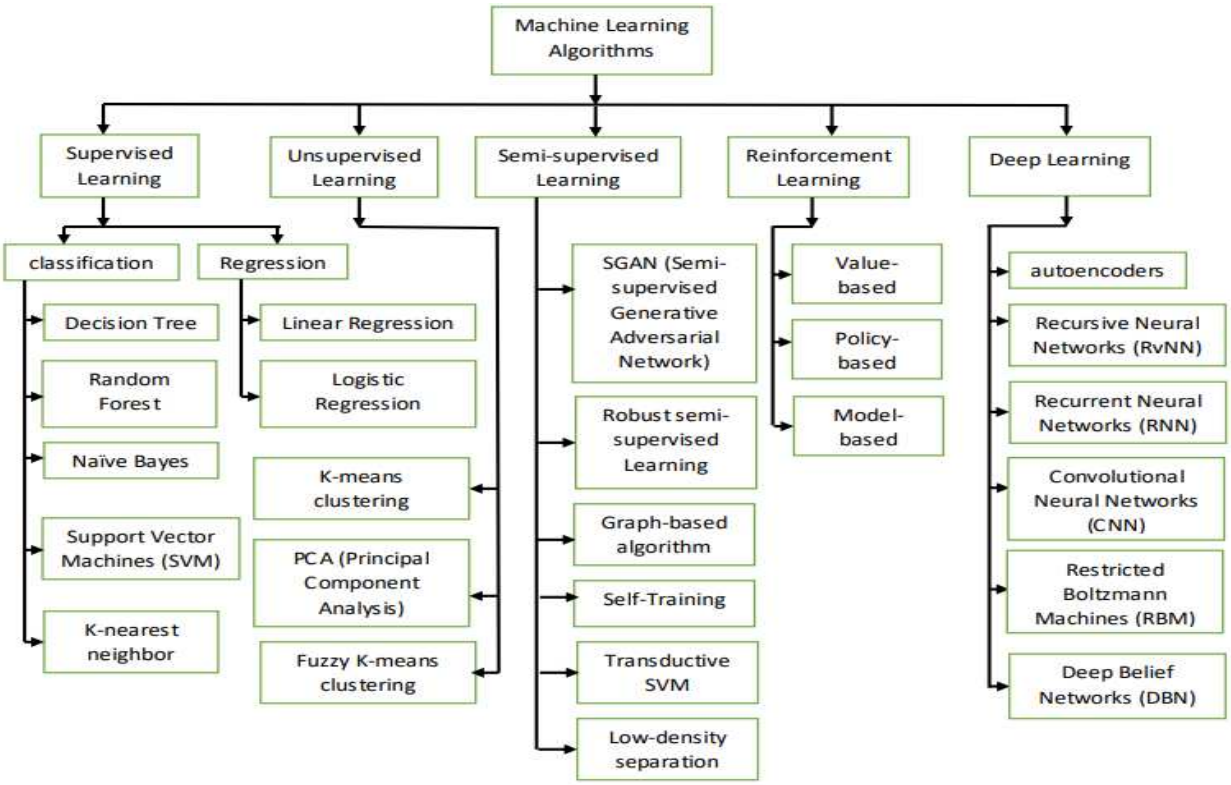

Figure 5. Classification of Machine Learning algorithms that can be implemented in web attacks detection models

statistical inference, they also incorporated threat intelligence model, and reinforcement learning to protect against any type of XSS attack. This is so because according to their research applying only genetic algorithm to detect XSS attack results in many false positives. XSS vulnerabilities can be analyzed using one of the three analysis approaches namely static analyses, dynamic analyses, and hybrid analysis. There has been a growing trend in using machine learning techniques to detect and prevent XSS attacks, however, they are not effective to novel and heterogeneous XSS attacks.

F. Survey on Metaheuristic Techniques (2018)

In [8], the researchers proposed a model by incorporating a Genetic Algorithm, which is metaheuristic approach, for XSS detection. Therefore, I read few papers on metaheuristic approaches that can be applied with machine learning models for XSS detection or prevention. One of the best papers in my perspective was the one written by Mohamed Abdel-Basset et al. in 2018 [9]. They presented a comprehensive survey of metaheuristic techniques, variants and taxonomies. According to [9], metaheuristics is a black box searching technique that can be applied to almost all optimization problems. These techniques are generally nature inspired (genetic algorithms, Clonal Selection algorithm), non-nature inspired, bio-inspired, trajectory-based, and population-based. The researchers added a new classification: metaphor based and non-metaphor based. The former simulates the natural phenomena or human behavior in real life while the later represents the techniques that are not related to any natural process. To clarify, the metaphor based classification covers the models based upon biology, physics, swarm, social, music, chemistry, sport and math. Whereas, the non-metaphor based classification depends upon the usage of memory, neighborhood structures, and objective function. Adaptive metaheuristics, chaotic metaheuristics and metaheuristics acceleration are some of the variants of metaheuristics discussed in the chapter. Although metaheuristics cannot solve all problems, applying hybridization of metaheuristics can prove to be beneficial in such scenarios. The paper also discussed about the recent trend of utilizing parallelism in metaheuristics for solving dynamic or high dimensional real life optimization problems, for example, Parallel Genetic Algorithm. These parallel schemas can also be hybridized in order to improve the efficiency of the solution. In the end, 10 metaheuristic algorithms (Genetic Algorithm, Particle Swarm Optimization, Base Optimization Algorithm, Harmony Search, TLBO, Biogeography-Based Optimization, Flower pollination algorithm, Multi-Verse Optimizer, SCA, Whale Optimization Algorithm) were applied to solve a Weld Beam Design Problem. Out of the 10 applied algorithms, TLBO's performance is found to be superior than others. The non-parametric Friedman test is used to statistically analyze the experimental results.

G. Improved N-Gram Approach

Rui Wang et. al (2015) [10] proposed a machine learning approach inspired from n-gram model to detect XSS attacks in online social networks and named it as improved-n gram model. Their proposed model aims at extracting vectors' features (keywords, JavaScript, URL, HTML tags), creating its own classifier (using Alternating Decision Tree to avoid limitation of Naïve Bayes SVM), and finding hidden malwares using n-gram approach. Researchers claim that n-gram approach is suitable for 2 gram or 3 gram models. Since, XSS attacks spread at much faster rate than other attacks in social networks, the spread speed of suspicious HTML tags, suspicious URLs, and suspicious JavaScript strings are taken

into consideration. The data samples are marked as suspicious or non-suspicious depending upon the number of matched instances in two libraries (malicious library and benign library). Their approach achieved satisfactory performance in precision and recall.

H. Static Analysis Method for XSS detection

G. Usha et al. (2020) [11] proposed a static analysis and data mining method to remove XSS vulnerability. The method aims to detect the malicious links and remove them from the source code. Their method performs better than improved n-gram model [9]. Firstly, the paper discusses about the sub-categories of XSS attacks followed by risks and threats of XSS in brief. Secondly, it throws light on existing basic XSS defense mechanisms such as secure encoding and validation, filtering, disabling the injected script execution. ESAPI and AntiXSS libraries can be used to escape the unsafe characters in HTML. Pixy tool which is an open source software can be used to detect XSS vulnerabilities in PHP scripts. Thirdly, the paper gives overview of 8 existing XSS attack detection techniques. Finally, the propose architecture is discussed where it has three parts: the analyzer which analyze the code of the URL, XSSV detector that detects malicious links in the URL, and XSSV remover that removes the detected malicious links to protect the user data. The researchers used Java/JSP language, MySQL database, Weka tool for graph generation using k-fold cross validation technique (k is set to 10) in the implementation of their model.

I. Intrusion Detection System based on Machine Learning

Sushant Sharma et al. (2020) [12] have attempted to figure out the main cause of false positive and false negative results in machine learning based web-attacks detection models with their proposed feature set extraction technique. It is claimed that feature set extraction or feature engineering plays a vital role in machine learning models. In their experiment, the results concluded that J48 outperforms much better than other two classifiers with 94.5% true positive rate and 94.7% precision rate, while Naïve Bayes produced best result for F1-measure at 93.9%. The researchers extracted 20 features from the dataset. Moreover, the same has been reported in study performed by FarhadAlam and Sanjay Pachauri (2017) [13] about the performance of J48 classifier where a comparative study was performed on three classification algorithms namely, J48 Decision Tree, Naïve Bayes, One Rule using Weka tool to detect fraud cases in credit card transactions. In their study, it is concluded that the performance of J48 decision tree outperforms than the other two as the time required to construct the model is less, prediction accuracy is higher, and the number of correctly classified instances are also more.

J. RSMP- Using End-to-End Deep Learning with intrusion Detection

Yao Pan et al. (2019) [14] stated that traditional intrusion detection systems often rely on supervised machine learning to differentiate between normal and abnormal traffic. This requires large set of training data which is costly, erroneous and time consuming. In addition to this, intrusion detection methods to defense against web attacks often requires in-depth knowledge of the security domain. Furthermore, existing unsupervised machine learning approach requires manual feature selection and also results in false positive rates. Moreover, attack detection using static and dynamic analysis is not effective in detecting and removing web vulnerabilities. In their paper, the researchers have claimed that deep learning can effectively performs classification and feature selection automatically. Their approach is based upon RSMT (Robust Software Modelling Tool) to detect different types of attacks having distinct characteristics. XSS attack is detected using stacked denoising autoencoder which is a symmetric deep neural network.

K. Supervised Machine Learning Approach to Vulnerability detection in OAuthentication protocol

K. Munonye and M. Peter (2021) in [16] proposed a supervised machine learning approach in combination with Finite State machine (FSM) and fuzzing algorithm for the detection of web vulnerability in OAuth 2.0 protocol. In collaboration with OpenID Connect service, it allows authorization and authentication of users' identity on other third party websites via existing sign on options such as Facebook, Google, Yahoo, etc. Mitigating vulnerability in OAuth protocol would assist in preventing cross-site scripting attacks. The approach leveraged by the researchers is pretty simple. Any modification of the HTTP parameters' values between the first phase and final phase resulting in undefined output could possibly represent presence of vulnerable code in the system. Features of OAuth were extracted using Principle Component Analysis and Factor analysis. Performance of model was evaluated using seven different classifiers to identify the best results with their mapped dataset. Out of the seven classifiers, Gradient Boosting Classifier performed excellent producing 83% accuracy, 96% precision rate,

V. CONCLUSION

After going through several research and survey papers on XSS attack detection techniques, it can be concluded that while vast research has been done in the field of supervised and unsupervised machine learning detection models, there is lack of research conducted in semi-supervised, reinforcement, and deep learning neural networks. Furthermore, metaheuristic algorithms can prove to be effective in performance evaluation of the web attack detection models. Please refer to figure 5 for sub-categories of machine learning algorithms that can be implemented with web attacks mitigation and detection models. It is recommended that instead of using single algorithms for feature extraction and attack detection, hybrid techniques must be used. Last but not the least, researchers should implement different types of classifiers with their unique dataset in order to receive better outcomes and accuracy score.

ACKNOWLEDGMENT

I am extremely grateful to the entire research community for guiding me to the right direction of my research area and comprehending the concepts related to web application security through research papers, survey papers, online articles and research blogs. Furthermore, I would like to thank Prof. Vikas Wasson and Dr. Surender Singh Dahiya for their kind support in identifying research challenges and providing the right direction to my research work. Finally, I would like

to thank my friends and family members for careful reading of my manuscript and their insightful comments and suggestions.

REFERENCES

[1] Alon Boxiner, Eran Vaknin, Alexey Volodin, Dikla Barda, Roman Zaikin, "Tik or Tok? Is TikTok secure enough?" Tik or Tok? Is TikTok secure enough? - Check Point Research, 8 January 2020.

[2] Heiderich, M., Jörg Schwenk, Tilman Frosch, Jonas Magazinius and Edward Z. Yang. "mXSS attacks: attacking well-secured web-applications by using innerHTML mutations." Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security (2013): n. pag.

[3] Stefano Calzavara et al., Machine Learning for Web Vulnerability Detection: The Case of Cross-Site Request Forgery, 22 January 2020, Co-published by the IEEE Computer and Reliability Society, pp. 1- 10

[4] K. Masashi et al., Exploring the Dark Web for Cyber Threat Intelligence using Machine Leaning, IEEE 2019, pp. 200-202

[5] Fang, Y.; Li, Y.; Liu, L., "Deepxss: Cross site scripting detection based on deep learning". In Proceedings of the 2018 International Conference on Computing and Artificial Intelligence, Chengdu, China, 12–14 March 2018; pp. 47–51.

[6] Yong Fang et al., "RLXSS: Optimizing XSS Detection Model to Defend Against Adversarial Attacks Based on Reinforcement Learning", Future Internet 2019, pp. 1-13.

[7] Kaur, G., Malik, Y., Samuel, H.W., & Jaafar, F. "Detecting Blind Cross-Site Scripting Attacks Using Machine Learning," SPML '18, 2018

[8] Iram Tariq, Muddassar Azam Sindhu, Rabeeh Ayaz Abbasi, Akmal Saeed Khattak, Onaiza Maqbool, Ghazanfar Farooq Siddiqui (2021). Resolving Cross-site Scripting Attacks through Genetic Algorithm and Reinforcement Learning, Volume 168, 15 April 2021, Article number 114386, Published by Elsevier Ltd.

[9] Mohamed Abdel-Basset, Laila Abdel-Fatah, Arun Kumar Sangaiah (2018). Chapter 10 – Metaheuristic Algorithms: A Comprehensive Review, In Intelligent Data-Centric Systems, Computational Intelligence for Multimedia Big Data on the Cloud with Engineering Applications, Academic Press, 2018, 185-231, ISBN 9780128133149.

[10] Wang R. Jia X. Li Q. Zhang D., "Improved N-gram Approach for Cross-site Scripting Detection in Online Social Network," 2015 Science and Information Conference (SAI) IEEE explore, pp. 1206-1212, 2015

[11] Usha G., Kannimuthu S., Mahendiran P.D., Shanker A.K. and Venugopal D. "Static analysis method for detecting cross site scripting vulnerabilities", Int. J. Information and Computer Security, Vol. 13, No. 1, 32–47,2020.

[12] Sharma, S., Zavarsky, P., & Butakov, S., "Machine Learning based Intrusion Detection System for Web-Based Attacks. 2020 IEEE 6th Intl Conference on Big Data Security on Cloud (BigDataSecurity)," IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS),pp. 227-230, 2020.

[13] Alam, F., & Pachauri, S., "Comparative Study of J 48, Naive Bayes and OneR Classification Technique for Credit Card Fraud Detection using WEKA,"

[14] Pan, Y., Sun, F., Teng, Z., White, J., Schmidt, D., Staples, J., & Krause, L. (2019). Detecting web attacks with end-to-end deep learning. Journal of Internet Services and Applications, 10, 1-22.

[15] Chen, X., Mohan Li, Yu Jiang and Yanbin Sun. "A Comparison of Machine Learning Algorithms for Detecting XSS Attacks." ICAIS (2019),pp.214-224.

[16] Munonye, K., Péter, M.., "Machine learning approach to vulnerability detection in OAuth 2.0 authentication and authorization flow". International Journal of Information Security, 13 May,2021.

[17] Chen, Y., Santosa, A., Yi, A.M., Sharma, A., Sharma, A., & Lo, D. (2020), "A Machine Learning Approach for Vulnerability Curation". Proceedings of the 17th International Conference on Mining Software Repositories, June 2020, pp. 32-42