

A Project Report on DETECTION OF MALICIOUS ATTACKS ON WEBSITES AND WEB SERVERS USING MACHINE LEARNING TECHNIQUES

**Bachelor of Technology
IN
COMPUTER ENGINEERING**

BY

**Gauhar Ayub Khan
Enrolment Number: GK9110**

**Mohiuddeen Khan
Enrolment Number: GK9121**

Under the Guidance of

**Dr. Nadeem Akhtar
Department of Computer Engineering
Zakir Husain College of Engineering & Technology
Aligarh Muslim University
Aligarh (India)-202002
April 2022**



Dated 15/04/2022

Declaration

The work presented in the project entitle "Detection of Malicious Attacks on Websites and Web Servers Using Machine Learning Techniques" was submitted to the Department of Computer Engineering, Zakir Husain College of Engineering and Technology, Aligarh Muslim University Aligarh, for the award of the degree of Bachelor of Technology in Computer Engineering, during the session 2021-22, is my original work. I have neither plagiarized nor submitted the same work for the award of any degree.

Date: 15/04/2022

A handwritten signature in blue ink, appearing to read "Gauhar", is placed over a rectangular stamp.

(Signature)

Gauhar Ayub Khan

Place: Aligarh

A handwritten signature in blue ink, appearing to read "M Khan", is placed over a rectangular stamp.

(Signature)

Mohiuddeen Khan



Dated 15/04/2022

Certificate

This is to certify that the Project Report entitled "Detection of Malicious Attacks on Websites and Web Servers Using Machine Learning Techniques", being submitted by "Gauhar Ayub Khan & Mohiuddeen Khan", in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Engineering, during the session 2021-22, in the Department of Computer Engineering, Zakir Husain College of Engineering and Technology, Aligarh Muslim University Aligarh, is a record of candidate's own work carried out by him under my (our) supervision and guidance.

Dr. Nadeem Akhtar

Assistant Professor

Department of Computer Engineering
ZHCET, AMU, Aligarh

Table of Contents

1.0 Title	3
2.0 Abstract	4
3.0 Acknowledgement	5
4.0 Introduction	6
5.0 Literature Review	7
6.0 Objectives	13
7.0 Methods and Methodologies	14
8.0 Our Approach	16
9.0 Output	18
10.0 Results	20
11.0 Conclusion and Future Work	21
12.0 Limitations	22
13.0 References	23

DETECTION OF MALICIOUS ATTACKS ON WEBSITES AND WEB SERVERS USING MACHINE LEARNING TECHNIQUES

Using machine learning techniques, we detected malicious attacks on websites and web servers. We focussed primarily on detecting XSS attacks because they are the most common. To create our system, we used a modified ensemble learning approach.

ABSTRACT

Web applications are used everywhere and deal with sensitive and personal information. As a result, they are a target for malware that exploits vulnerabilities to gain access to unauthorised data stored on the computer. SQL injection, cross-site scripting (XSS), and other attacks fall into this category. XSS attacks can harm the victim by stealing cookies, modifying a web page, capturing clipboard contents, keylogging, port scanning, dynamic downloads, and other methods. As a result, web application security is a critical task for developers. The most common security flaw in web applications is the failure to validate client input or the environment, and such flaws are frequently discovered and exploited on both the client and server sides. SQL injection and XSS remain among the top ten OWASP vulnerabilities.

In this project, we looked into using machine learning techniques to create classifiers that can detect XSS in JavaScript. We concentrated on stored or persistent XSS, which occurs when a malicious script is injected into a web application and stored in the database. An attack of this type could target blogs, forums, comments, or profiles.

ACKNOWLEDGEMENT

We would like to express our heartfelt gratitude to Dr Nadeem Akhtar, our project supervisor, whose invaluable assistance, comments, and recommendations helped us complete this project and make it a complete success. His ideas and suggestions had been critical to the project's success.

We'd also like to thank our parents and friends for providing us with invaluable advice and recommendations throughout the process.

Dr Nadeem Akhtar

Mohiuddeen Khan

Gauhar Ayub Khan

Date: 15 April 2022

INTRODUCTION

Web applications are used all over the place and deal with sensitive and personal data. As a result, they are a prime target for malware that exploits vulnerabilities to gain access to unauthorised data on the computer.

SQL injection, cross-site scripting (XSS), and other attacks are examples of this type. XSS attacks can cause harm to the victim by stealing cookies, modifying a web page, capturing clipboard contents, keylogging, port scanning, dynamic downloads, and other techniques.

According to a recent Precise Security study, the most common cyberattack is the XSS attack, which accounts for roughly 40% of all attacks. Despite being the most common, the majority of these attacks are simple and carried out by inexperienced cybercriminals.

Cross-site scripting vulnerabilities have varying degrees of impact depending on the web application. Session hijacking, credential theft, and other security flaws are all included. By exploiting a cross-site scripting vulnerability, an attacker can impersonate a legitimate user and take over their account. Many techniques have been used to protect websites, web servers, and data from these attacks, including static and dynamic analysis of scripts and code, but they are ineffective, so we need a system that can detect malicious scripts.

LITERATURE REVIEW

BACKGROUND:

Cross-site scripting (XSS) is one of the most common attacks used to annoy the user, steal or modify sensitive data, expose files, install malicious programmes, and so on. There are three types of XSS attacks that have been identified and studied. In Stored XSS, the attacker injects malicious code, most commonly by leaving comments on a blog, registering on a website, or filling out web forms. When a victim visits a web page that uses the infected web service, the malicious code is persistently stored in a database and executed.

In most cases, the attacker sends an email to the victim with a link to an infected web service or website. When a user accesses an infected web service or website, malicious code is included in the HTTP response, activated, and sensitive data is sent to the attacker. The malicious code is injected into the DOM element and modifies the DOM in the victim's web browser.

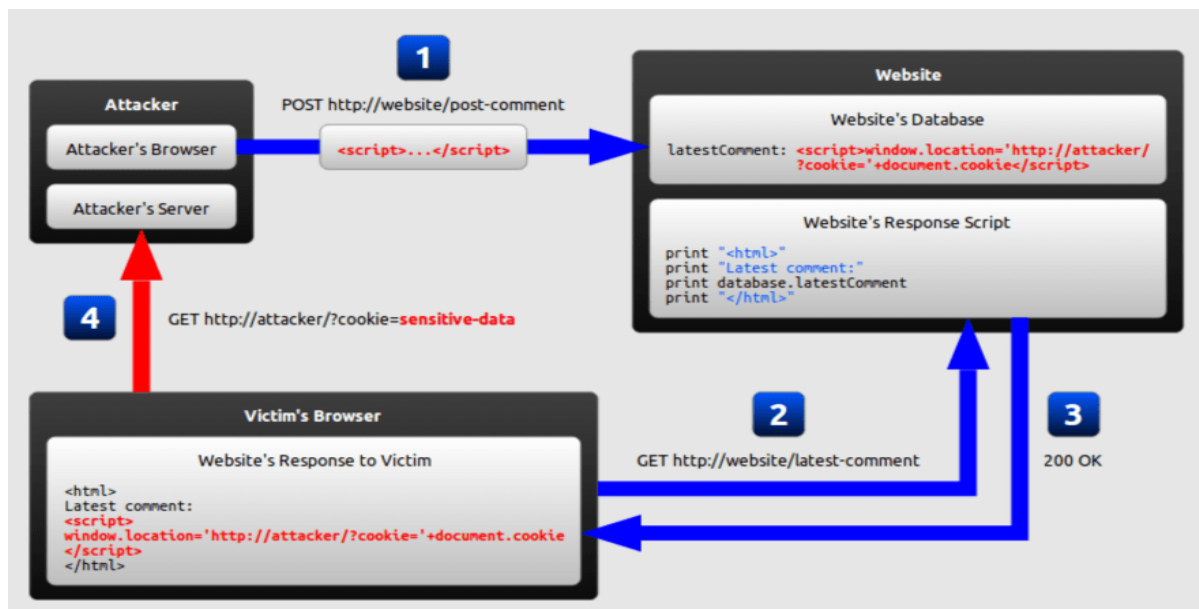


Fig. XSS Attack FlowChart [13]

EXISTING APPROACHES:

A number of approaches have been taken to dealing with attacks on websites.

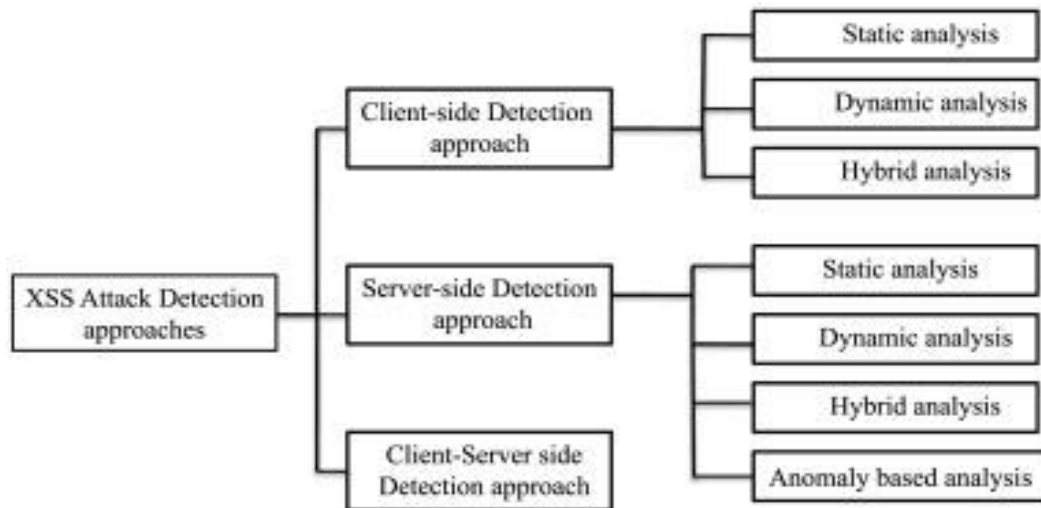


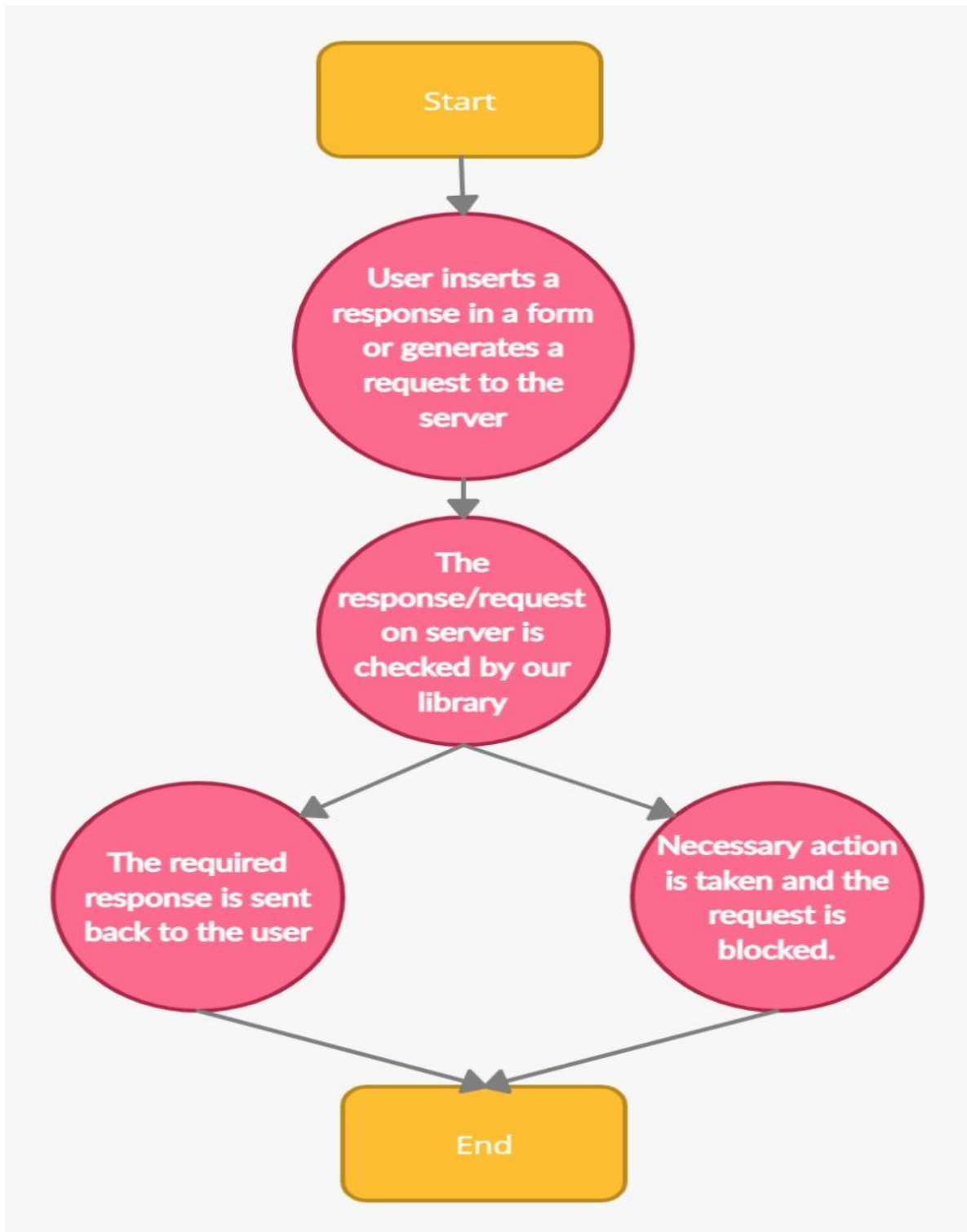
Fig. XSS Detection Existing methods[6]

- The standard approach for the web application developer is to use sanitizing and escaping to prevent untrusted content from being interpreted as code.
- Alternatively, parser-level isolation can confine user input data during the lifetime of the web application.

- Another technique to defend against XSS vulnerability is to use randomized namespace prefixes with primitive markup language elements to make it hard for the attacker to use these elements.
- Previous methods aim to remove malicious elements from untrusted data, however, as with blacklists some XSS vectors can easily bypass many powerful filters.
- Machine learning techniques have been applied to detecting XSS attacks and are attractive because they can adapt to changes and variations in malicious scripts.

RELATING OUR WORK:

Our goal is to use machine learning techniques to test and compare code from a variety of sources, such as forms, to determine if it is a dangerous script or not. We'll put the basic idea to the test by building our own server and targeting it in various ways.



DATASET:

The first data set was simple text data with labels 0 or 1 i.e XSS attack codes and non-malicious code respectively. This data set was used for the NLP model.

	B	C
	Sentence	Label
0	test</tt>	1
2	 Steering for the 1995 "<a href="/wiki/History_of_autonomous_cars#19	0
3	 <cite class="citation web"><a rel="nofollow" class="external text" href	0
4	. doi :<a rel="nofollow" c	0
5	<li id="cite_note-118">^ 	0
6	Contextualism 	0
7	<li id="cite_note-Representing_causation-95">^ <a href="#cite_ref-Represent	0
8	<tr><td class="plainlist" style="padding:0 0.1em 0.4em">	0
9		0
10	Mind–body problem 	0
11	<input autofocus>	1
12	<col draggable="true" ondragenter="alert(1)">test</col>	1
13	<caption onpointerdown=alert(1)>XSS</caption>	1

The second dataset was an integer dataset with 46 features and with labels 0 or 1 i.e XSS attack codes and non-malicious code respectively. This data set was used to train the Logistic Regression and the SVM model. This data set was generated from dataset 1 using python script.

	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ	AR	AS	AT	AU
1	Contains Less Than	Contains Greater Than	Contains Location	Contains Search	Contains &#	Contains Colon	Contains Dots	Contains Space	Contains Quot	Contains Double Slash	Contains Vertical	Contains Bracket	Contains Alert	Class
2	1	1	0	0	0	1	1	0	0	1	0	0	1	1
3	1	1	0	1	0	1	1	0	0	1	0	0	1	1
4	0	0	0	0	0	1	1	1	0	0	0	0	0	0
5	1	1	0	0	0	1	1	0	0	1	0	0	1	1
6	0	0	0	0	0	0	1	0	0	0	0	0	0	0
7	0	0	0	0	0	0	1	1	0	0	0	0	0	0
8	0	0	0	0	0	0	0	1	0	0	0	0	0	0
9	0	0	0	0	0	0	0	1	0	0	0	0	0	0
10	0	0	0	0	0	0	1	1	0	0	0	0	0	0
11	1	1	0	0	0	1	1	1	1	1	1	0	0	0
12	0	0	0	0	0	1	1	0	0	1	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	1	1	0	0	1	0	0	0	0
15	0	0	0	0	0	1	1	0	0	1	0	0	0	0
16	0	0	0	0	0	0	1	0	0	0	0	0	0	0
17	1	1	0	0	0	1	1	1	1	1	0	0	1	1
18	0	0	0	0	0	0	1	0	0	0	0	0	0	0
19	0	0	0	0	0	1	1	1	1	1	0	0	0	0
20	0	0	0	0	0	1	1	0	0	1	0	0	0	0
21	0	0	0	0	0	0	1	0	0	0	0	0	0	0
22	0	0	0	0	0	1	1	0	0	1	0	0	0	0
23	0	0	0	0	0	0	1	1	0	0	0	0	0	0
24	0	0	0	1	0	0	1	1	0	0	0	0	0	0
25	0	0	0	0	0	0	1	0	0	0	0	0	0	0
26	0	0	0	0	0	0	1	1	0	0	0	0	0	0

In our dataset, we have malicious scripts in one column and the other column has boolean values that define whether the script is malicious (1) or non-malicious(0).

OBJECTIVES

Our goal is to use machine learning methods to test and compare code from various sources, such as forms, to determine whether or not it is a dangerous script. We'll put the original concept to the test by constructing our own server and attacking it in various ways.

We will conduct preliminary research into XSS attacks, and after developing a robust model for the same, we will apply our models to various types of attacks in future. We will effectively turn this tool into a library, contributing to the field of computer and network security by allowing anyone to secure their website or network.

METHODS AND METHODOLOGIES

METHODOLOGIES USED:-

Ensemble learning: Ensemble learning is the process of systematically generating and combining many models, such as classifiers or experts, to tackle a specific computational intelligence issue. Ensemble learning is largely used to improve a model's performance (classification, prediction, function approximation, etc.) or to lessen the risk of an unintentional poor model selection. Ensemble learning may also be used to provide a confidence level to the model's choice, pick optimal (or near ideal) features, data fusion, incremental learning, nonstationary learning, and error-correcting. The figure below shows ensemble learning:-

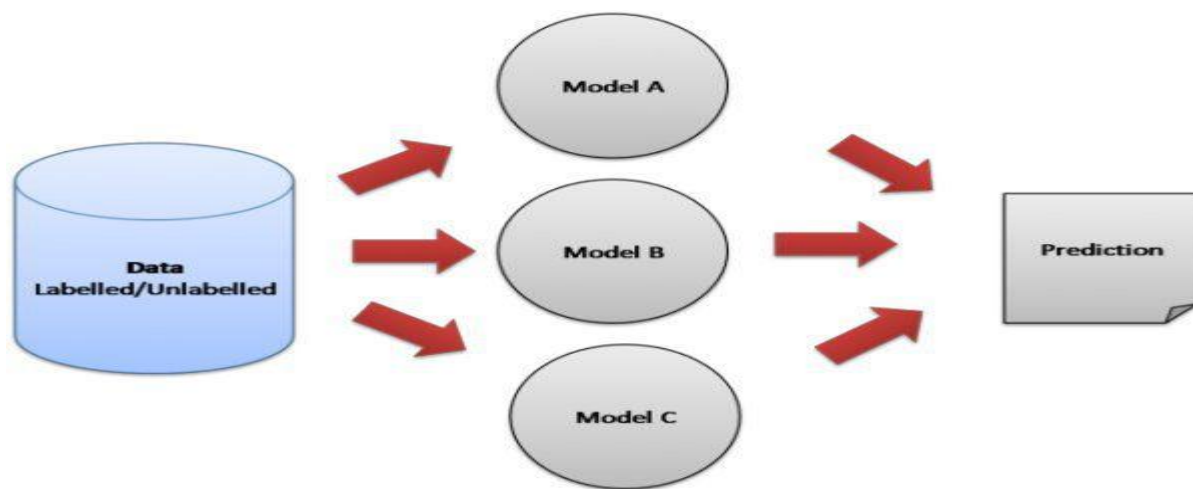


Fig. Ensemble learning[6]

Bootstrap aggregating: Bagging, also known as Bootstrap aggregating, is an ensemble learning strategy that helps machine learning algorithms increase their performance and accuracy. It decreases the variance of a prediction model by dealing with bias-variance trade-offs. Bagging is a technique for avoiding data

overfitting and is used in both regression and classification models, as well as decision tree methods.

Neural networks: A computer learning system that employs a network of functions to comprehend and transform a data input in one form into the desired output, generally in another form, is known as an artificial neural network learning algorithm, or neural network, or simply neural net. Human biology and the way neurons in the human brain work together to interpret inputs from human senses inspired the artificial neural network concept. Machine learning algorithms employ a variety of tools and methodologies, including neural networks. The neural network may be used as a component in a variety of machine learning techniques to convert complicated data into a format that computers can comprehend.

Logistic regression: Logistic regression is a type of supervised learning that uses a logistic/sigmoid function to estimate probabilities and assess the association between a categorical dependent variable and one or more independent variables. Despite its name, logistic regression is not employed for regression problems in which the goal is to predict real-valued outcomes. It's a classification issue in which a collection of independent factors is utilized to predict a binary result (1/0, -1/1, True/False).

Our proposed approach is shown in the figure below. Firstly, we will take our data from Kaggle which is nothing but codes and their classification(Malicious code/Non-malicious). The code however is in the text format and can only be used to train the NLP model for classification.

OUR APPROACH

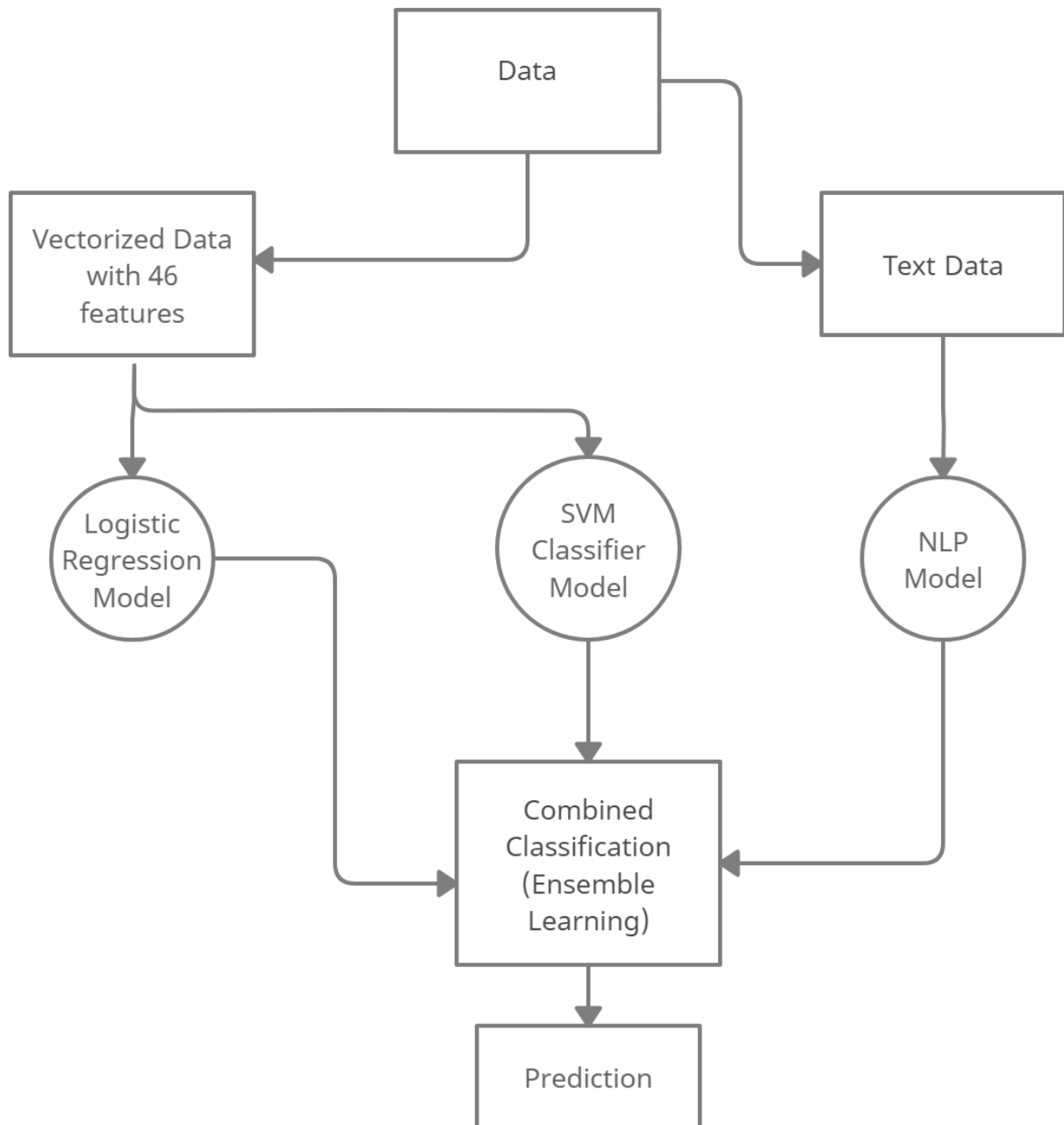


Fig. Our proposed ensemble learning approach

We propose an ensemble model containing 3 models (more models will be added when the first version of the project is finished). We chose 3 models to increase the robustness of the overall result. The first model is the NLP model to which we will directly feed our data containing codes and their results.

After that, we are going to make a script to create data containing an integer and real data from the code. For example:- We will count the number of special keywords used in the malicious and non-malicious code and this will be one attribute of the training data. Similarly, the next attribute will be the number of lines of the code and so on. This way we will generate real data ready to be fed into logistic regression and neural nets.

After taking results from all the 3 models, we will classify the code as malicious or non-malicious after hard voting. For ex:- Suppose the outputs of the 3 models are:-

NLP model - 1 (malicious)

Neural network - 0 (non-malicious)

Logistic regression model- 0 (non-malicious)

Since 2 models out of 3 predict the code as non-malicious, the code will be classified as **non-malicious**.

OUTPUT

PAYMENT BILL RECORD

Payment amount

\$ 2000.00

Name

mohiuddeen

Address


2546, street1

Occupation

student

Remarks

<nextid onmousemove=""alert(1)"">test</nextid>



Submit

```

PS E:\Documents\major_project\app\xss-attack-ml> cd test_server
PS E:\Documents\major_project\app\xss-attack-ml\test_server> node server.js
Express server listening on 8500
mohiuddeen
2546, street1
student
<nextid onmouseover=""alert(1)"">test</nextid>
1
Form data contains malicious XSS script

```

```

PS E:\Documents\major_project\app\xss-attack-ml\flask_ml_server> python main.py
SVM result = [0]
Logistic result = [0]
Given string is safe
nlp result = [0]
SVM result = [0]
Logistic result = [0]
Given string is safe
nlp result = [0]
SVM result = [0]
Logistic result = [0]
Given string is safe
nlp result = [1]
SVM result = [1]
Logistic result = [1]
Given string is an XSS string
127.0.0.1 - - [15/Apr/2022 23:25:51] "POST /predict HTTP/1.1" 200 -

```

RESULTS & METRICS

NLP:

Train Accuracy -> 99.87 %

Test Accuracy -> 99.76 %

LOGISTIC REGRESSION:

Train Accuracy -> 98.56 %

Test Accuracy -> 98.26 %

SUPPORT VECTOR MACHINE:

Train Accuracy -> 98.79 %

Test Accuracy -> 98.57 %

ENSEMBLE MODEL:

Train Accuracy -> 98.87%

Test Accuracy -> 98.93%

CONCLUSION AND FUTURE WORK

We will be extending our system to detect more attacks and we will also work to improve its accuracy. We will also make this library robust and scalable so that it can be integrated easily within various applications.

As websites contain very important and secretive information, it is required to have a software or library apart from a firewall that can stop and counter these attacks. We are able to improve the security of web servers by building a library to check and counter threats. We'll be able to prevent harmful scripts from infecting websites. We can further extend this project to work on different kinds of attacks like SQL injection, XXE etc.

LIMITATIONS

- Since this is the first version of our project, our system is currently limited to detecting XSS attacks.
- The proposed algorithm is not optimized well; since the request is sent back to the user, using complex machine learning libraries will slow down the response.
- After a certain amount of attacks, the attacker may get aware of the attack identifying technology and may develop a new system to counter it.

REFERENCES

1. Fawaz A. Mereani and Jacob M. Howe. 2018. Detecting Cross-Site Scripting Attacks using Machine Learning. In: Hassanien A., Tolba M., Elhoseny M., Mostafa M. (eds). The International Conference on Advanced Machine Learning Technologies and Applications (AMLTA2018). AMLTA 2018. Advances in Intelligent Systems and Computing. 723. Springer, Cham. http://doi-org-443.webvpn.fjmu.edu.cn/10.1007/978-3-319-74690-6_20.
2. Detection of XSS Attack and Defense of REST Web Service – Machine Learning Perspective Malinka Ivanova Anna Rozeva ICMLSC'21: 2021 The 5th International Conference on Machine Learning and Soft Computing January 2021 Pages 22–28 <https://doi.org/10.1145/3453800.3453805>.
3. Kirda, E., Kruegel, C., Vigna, G., Jovanovic, N.: Noxes: a client-side solution for mitigating cross-site scripting attacks. In: Symposium on Applied Computing, pp. 330–337. ACM Press (2006).
4. Su, Z., Wassermann, G.: The essence of command injection attacks in web applications ACM SIGPLAN Not. 41(1), 372–382 (2006).
5. https://en.wikipedia.org/wiki/Ensemble_learning
6. <https://www.kdnuggets.com/2020/03/making-sense-ensemble-learning-techniques.html>
7. <https://www.sciencedirect.com/science/article/abs/pii/S1084804518302042>
8. R. Polikar, "Ensemble learning," in Ensemble machine learning, Springer, 2012, pp. 1–34.

9. G. E. Rodríguez, J. G. Torres, P. Flores, and D. E. Benavides, "Cross-site scripting (XSS) attacks and mitigation: A survey," *Comput. Networks*, vol. 166, p. 106960, 2020.
10. S. Akaishi and R. Uda, "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)*, 2019, pp. 1–6.
11. <https://owasp.org/www-community/attacks/xss/>
12. I. Hydera, A. B. M. Sultan, H. Zulzalil, and N. Admodisastro, "Current state of research on cross-site scripting (XSS) - A systematic literature review," *Inf. Softw. Technol.*, vol. 58, no. July 2015, pp. 170–186, 2015.
13. <https://www.indusface.com/blog/xss-examples-prevent/>
14. M. A. Ahmed, and F. Ali, "Multiple-path testing for cross site scripting using genetic algorithms," *Journal of Systems Architecture*, vol. 64, pp. 50-62, 2016.
15. Dr. G. Rama Koteswara Rao, K.V.J.S. Sree Ram, M. Akhil Kumar, R. Supritha, and S. Ashfaq Reza. 2017. Cross site scripting attacks and preventive measures. *International Research Journal of Engineering and Technology*. 4(2). 2016 – 2019.
16. https://www.researchgate.net/figure/The-Bagging-Bootstrap-Aggregation-scheme_fig2_284156704