National College of Ireland

Cross Site Scripting detection using Random Forest Bagging and Dataset Ensemble Modelling

MSc Internship
Cybersecurity

Shreyas Sudhir Barde
Student ID: x18198350

School of Computing
National College of Ireland

Supervisor: Niall Heffernan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Shreyas Sudhir Barde
Student ID:	x18198350
Programme:	MSc Cybersecurity **Year:** 2019-20
Module:	MSc Internship
Supervisor:	Niall Heffernan
Submission Due Date:	17/08/2020
Project Title:	Cross Site Scripting detection using Random Forest and Dataset Ensemble Modelling.
Word Count:	5688 **Page Count:** 17

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.
I agree to an electronic copy of my thesis being made publicly available on NORMA the National College of Ireland's Institutional Repository for consultation.

Signature: Shreyas Sudhir Barde

Date: 16/08/2020

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Cross Site Scripting detection using Random Forest Bagging and Dataset Ensemble Modelling.

Shreyas Sudhir Barde
X18198350

Abstract

Cross-site scripting known as XSS attack is a type of the most vulnerable and critical attack on web apps. Conventional strategies of detection of XSS are basically focused on the vulnerability of apps only, which are depending on the static and dynamic analysis. These methods seem frail in protecting applications from the wave of different sorts of payload attacks. In this study, the cross-site scripting detection methodology is introduced which is based on a dataset ensemble learning technic that utilizes different XSS datasets. This study has included ensembled dataset of real-world payloads which helps to accurate detection of real-time attacks. Along with that, this study is proposed a novel approach of the feature extractions from the malicious scripts, which leads to the exact detection of XSS attacks on the system. The outcomes of this study are reasonable and explainable. To reverify the results of the proposed approach, Parallel to the Random Forest model, Various other models have been tested on an ensembled real-world dataset of the XSS payloads. From the result of the proposed strategy, the accuracy and efficiency of this model can be clearly observed. Particularly when the attacking input is based on real-time payloads, this method and model detects any malicious script most accurately. Moreover, the use of a bagging algorithm improves the stability, accuracy, and reduces variance. It also helps to avoid overfitting of the model used for research. The accuracy has been observed with this model is 97.16% and the training time is 0.28 seconds.

1 Introduction

Web application security is being a foremost critical security issue because of the advancement in web application hacking tools and technologies [1]. As described in [2], XSS is among the top well-known vulnerabilities since a long time. Any malicious actor can take the remote access of any user's account and can easily manipulate critical information with the help of XSS vulnerabilities. In other words, a Cross-site scripting attack is another type of injection attack in which the hackers inject malicious XSS scripts in the webserver with the intent of exploitation. If input validations are not utilized properly while the development of web applications, the probability of XSS attacks execution in responding pages increases a lot. Many times, it has been observed that browsers execute such malicious scripts inside the user's machine and the hackers get access to the user's critical data like usernames and passwords [3].

In technical terms, XSS can cause data leaking as well as it allows hackers to take unauthorized access to any web application with the help of HTTP cookies. the escapement of particular characters limits to create and inject malicious script codes. This is one of the regularly used and successful countermeasures against the cross-site scripting attacks. There are few web developers who do not even understand the actual use of the escapement as well as the importance of the prevention of the XSS attacks. In addition, the XSS scripts can be injected in the web pages when vulnerability presents in a web server. Particularly defending an unknown intrusion with a newer vulnerability becomes a crucial task [4].

Existing studies are there for identifying malicious XSS scripts. Many times, Pattern matching strategies have been utilized for the detection of any cyber-attacks or detection of the malware. The same methods have been utilized for XSS detection. Moreover, for detecting such attacks, machine learning is being one of the most preferred areas in this field. Particularly, the Injection type of attacks can be detected by machine learning approaches with good accuracy [4].

However, there is a possibility that samples of malicious as well as benign scripts can be classified mistakenly in ML algorithms. Whereas benign or generous codes that don't incorporate the particular malevolent patterns should not be classified by pattern matching strategy ever. In the previous study carried out by the authors Yun Zhou and Peichao Wang, there is a large percentage of misclassified samples are present in the dataset they have utilized [5]. They have explicitly stated in their future work that the model they have implemented should need to test with much more different kinds of datasets and practical scenarios. Hence, this research is worked on and progressed on the pre-processing strategy for the training dataset with ensemble dataset modelling techniques. Along with that, this study has improved the accuracy by making changes in the feature extraction to detect the real-world attacks and payloads. Additionally, presented the use of a bagging algorithm for improvement in the stability, accuracy, and reduction in the variance which also helps to avoid overfitting of the model used in this research.

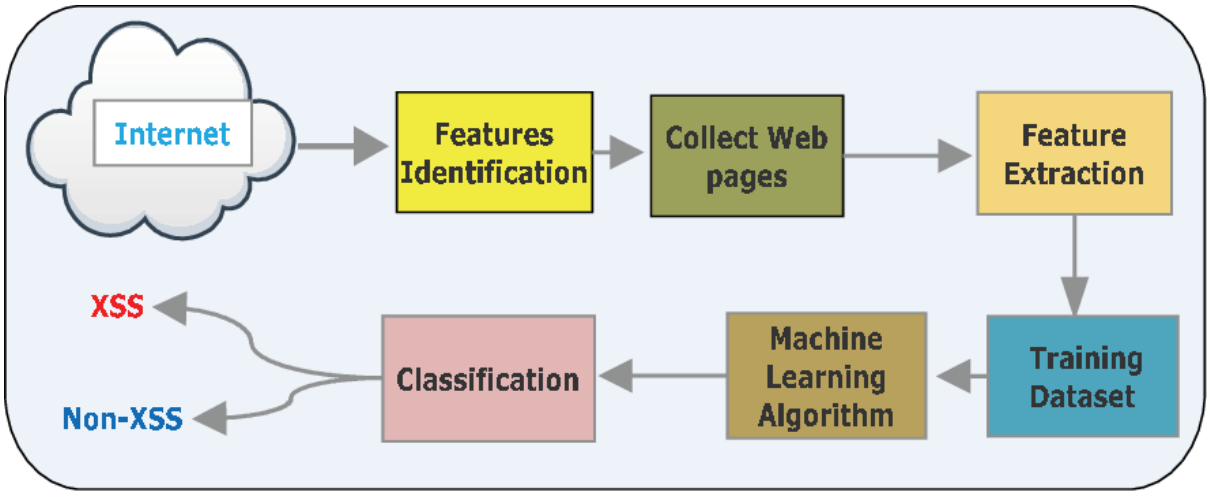

Figure 1: Conceptual representation of XSS detection using ML [2].

2 Related Work

In this modern era of information technology, Conventional strategies such as static and dynamic analysis are not appearing that successful to stop XSS attacks because of the expanding scope and availability of payloads of XSS. To overcome these difficulties, modern machine learning technics have been utilized in various research of the detection as well as prevention of XSS attacks. These studies accomplished great outcomes as well. This section presents previous investigations and studies carried out to detect and avoid XSS attacks.

2.1 Balanced and Ensembled Datasets for XSS detection using ML:

There are various Machine learning methodologies and models are present that have been utilized by past researchers for the prevention of XSS attacks. those methods and algorithms are astonishing as they can understand the traps and varieties of payloads utilized by hackers in any malicious script's codes. The study presented by Nunan et al., the segregation of the data samples in two classes i.e. XSS scripts and non-XSS scripts with the use of Naive Bayes algorithm and Support vector machine algorithm is carried out [6]. They have utilized various features for example length of the URL link, the count of domains used, the number of special characters as well as keywords used, and some of the HTML tags or JavaScript tags that are utilized. In this research, Weka has been utilized for the purpose of data classification and data pre-processing. The total malicious XSS script samples were approximately 15K. Moreover, around 57K benign scripts of Dmoz and 160K benign scripts of the ClueWeb dataset were utilized as samples. For assessing the model, the 10-fold cross-validation has been used and the average of that is calculated [6].

However, some issues can be observed in this research study. Firstly, the ratio of the malicious and the benign scripts in the dataset they have used. The malicious scripts included in the dataset were approximately 15K where the benign scripts were 158K. Hence it can be observed that the ratio of both scripts was approximately 1 : 10. Any dataset having such proportion is known as an imbalanced dataset. As stated in the study by He. et. al. [7], it is proved that the performance of the model reduces greatly if any imbalanced datasets are utilized. Along with this they explicitly mentioned that the penalty for any misclassified data should be balanced. Otherwise, the ratio of data should be adjusted while using such an imbalanced dataset [7]. Hence, it can be observed that in any case, Nunan et al. had not applied any such measures.

The difference can be decreased between the numbers of the majority script samples and the number of minority script samples using the Over-sampling method [8], [9]. This can be done by creating new instances such as minority script samples. Unlike under-sampling, In the Over-sampling method, every data sample can be saved and so do not lose important information. However, it is costly in terms of computing, particularly in tremendous imbalanced datasets. As discussed in [8] and [9], these are improved Over-sampling approaches that generate a model from minority samples by obtaining probability distribution.

Hence, for this study, an ensemble dataset has been developed using the combination of multiple techniques discussed in the above papers. In that dataset, multiple script types from

various cheat sheets and datasets have been used for accurate prediction of real-world attacks. A balanced dataset has been created and used with maintaining a good ratio of malicious and benign scripts from different datasets.

2.2 Classification of XSS scripts using Feature extraction:

The approach defined by Kaiho et al is based on the automatic classification of cross-site scripting payloads [10]. This research is centred on the characteristic of letters that are utilized in XSS scripts for the purpose of detection and filtration of the malicious XSS scripts from the benign or generous scripts. In one more research study by Matsuda et al., it has concluded that 32 different kinds of letters are regularly utilized by hackers in XSS payloads [11]. But Kaiho et al. observed that 32 among these 34 types of letters are usually utilized for normal user inputs as well. Hence, they used selected four types of letters only, which are surely was found in XSS payloads amid other types. Four sorts of letters are extricated from the overall input string and find out the assessment score of any input data. In that study, a total of 350 script payload samples are classified and with 5-fold tested there cross-validation. The greatest precision in this research noted was 98%. However, in this study, two issues can be observed. Firstly, the accuracy of classification can be a major problem here. Because they have shown the misclassification rate of 2% which is really high for real utilization. Moreover, it appears that the strategy they have implemented will not be that effective to detect unknown and recently made the latest XSS payloads. Because the old previous threshold from the previous studies has been utilized for the detection of modern complex payloads and hence the accuracy and precision of such classifications will definitely and will amazingly reduce.

Based on many studies, Keywords are imperative for recognizing XSS attacks. A few researchers concluded that the occurrence of the utilization of a few specific JavaScript methods in generous pages and in malicious pages is very diverse [12], for example, the eval JavaScript function usually utilized much more regularly for the scripting of malevolent pages as compared to benign pages. So that this research has included such suspicious function names as the keywords in this paper. In another study by Likarish et al. [13], he suggested 50 keywords in his paper.

Therefore, in this paper collective approach of previous studies along with some additional methods have been taken to improve the efficiency of the feature extractions in this paper.

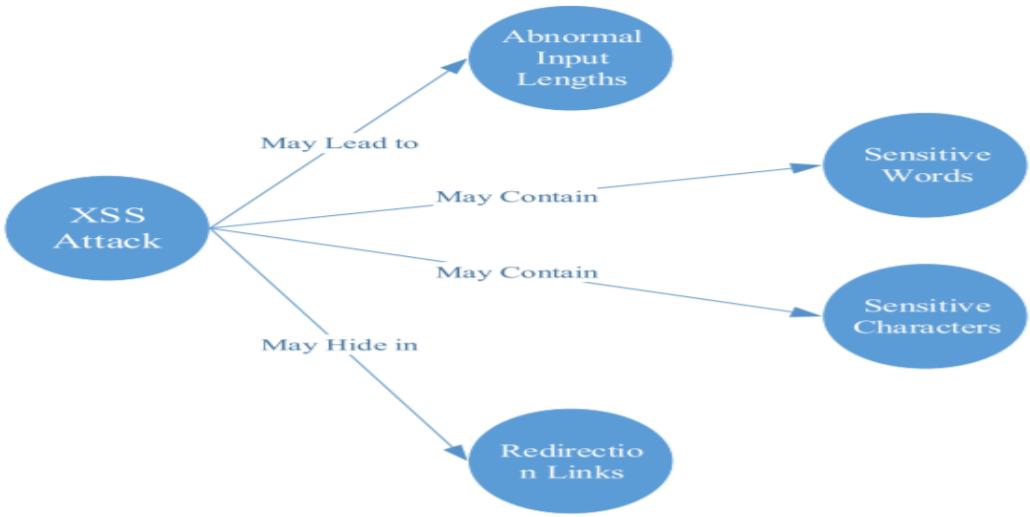

Figure 2: Ontology of XSS attacks [5].

2.3 Random Forest and Bagging Techniques for XSS detection:

Bagging is one of the most useful Machine learning technics generally used for the improvement in the stability, accuracy, and reduction in the variance which also helps to avoid overfitting of the model. The researcher Breiman presented the technique of bagging with the target of improving the results of classification of classifiers which are having an unstable base [14]. Basically, this bagging process is one of the methods of taking a random sample as a substitution of the same volume data from the actual first dataset. Hence, a few occurrences may show up multiple times whereas others may not show up not even a single time inside the updated or inherited training datasets. These are usually defined as bootstrap replicates [14].

There are multiple research-works that have been done focusing on bagging. An exploratory research study was proposed by Bhattacharyya and Nagi [15]. They achieved this with the utilization of 9 HD microarray datasets of cancer along with the 3 classifiers. These researchers presented a novel learning strategy in which they analyse class-specific precision of their strategy and compared them with every classifier and with bagging strategy as well. In the conclusion of their research, they stated that the classifiers bagging beats all the other learning strategies, even their own proposed strategy as well.

In one more research which was done by Gilbert and Tan [16], they used feature selection with the decision tree as a classifier to 7 different datasets. on top of that, they used bagging along with the decision tree. In conclusion, these authors mentioned that ensemble modelling methods with the bagging technics gave better results of classification. Chen et al. [17] also gave different bootstrapping approach in his research study. His work is focusing on the datasets which are not balanced. Researchers derived the actual balanced process of bootstrapping. He did this with the Random Forest algorithm. Because of this alteration, every decision tree got balanced bootstraps. This was done by a nondependent selection of the same number of major and minor instances.

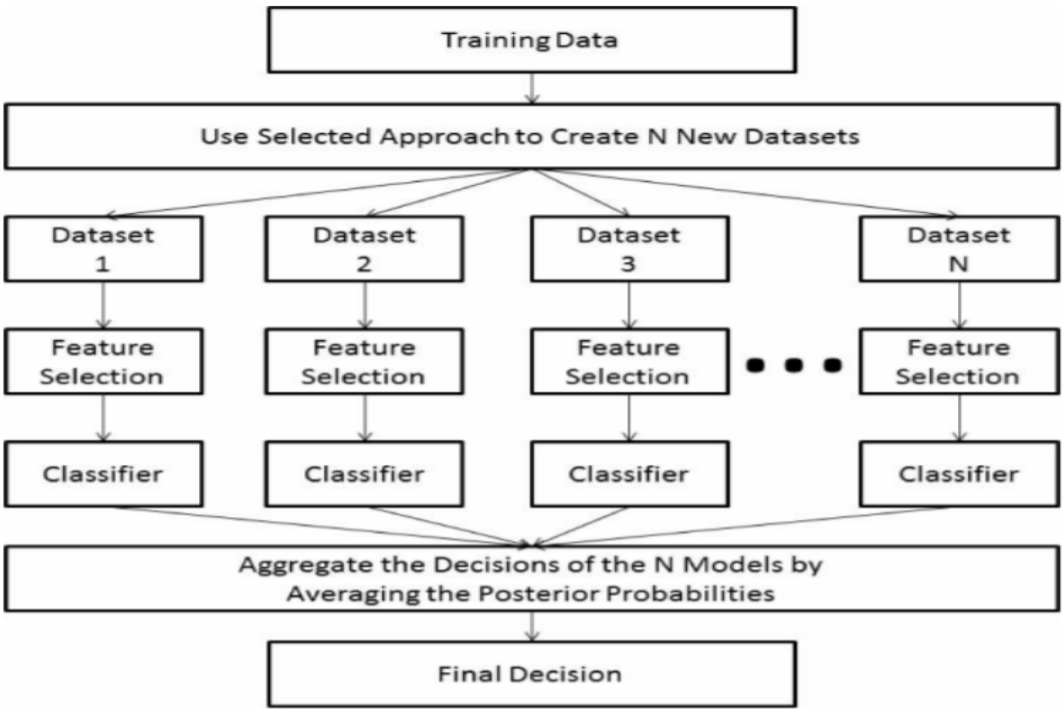

Figure 3: General select bagging algorithm [14].

After doing a critical analysis of previously mentioned research works, It can be observed that these works have some of their own inadequacies. For example, In the research study of Chen et al. [17], even if they have proposed the implementation of balanced bagging, datasets they used in this study are not enough as they are not high-dimensional sets. In the second research of Bhattacharyya and Nagi [15], Just the classic bootstrapping technic was utilized for the bagging method. Moreover, feature selection was also not applied. In the third study of Gilbert and While Tan [16], they used feature selection to 7 datasets along with that they applied bagging of the classifier of the decision tree. They did this to build models of classification. but for the bagging process of the classifier, the only classic bootstrapping methodology is utilized.
Considering all these previous works, Random Forest and Bagging Techniques for this model implementations has been used in this study and achieved an expected outcome with this approach and that is illustrated in detail in the next sections of this report.

3 Research Methodology

This section is focused on the followed research process for predicting knowledge in the data, utilizing machine-learning technics on datasets obtained from the past study as well as from some other sources. This section also highlights some important perspectives of Data Analytics and Data Mining.

In this research study, the KDD methodology has been followed as KDD methodology is the most recognized and widely used process for data mining [18]. The KDD stands for Knowledge Discovery in Databases. KDD is a wide process of obtaining particular knowledge in whole data. It focuses on the applicability of specific strategies of data mining. The overall objective of the KDD is to extricate knowledge among the whole data within the huge databases [18]. This does achieve by utilizing data-mining strategies to distinguish what is regarding information is. This can be done as per the details of measures as well as thresholds. Utilizing a database with the needed pre-processing and changes in that database is also a part of it [18]. This research has been conducted by following the general process of obtaining and rendering patterns from the data. That includes the reiterated execution of the steps given below:

3.1 STAGE ONE – DATA SELECTION:

In data selection stage of this research, an understanding of the domain of the application (i.e. XSS detection) has been developed based on the significant previous knowledge. Along with that the objectives of the end-user have also been taken into consideration while selecting a data. While the creation of a target dataset, more than two different datasets have taken into consideration. One dataset is from the Kaggle open dataset [19] where other datasets are from the GitHub open dataset projects [20]. While selecting a dataset for ensemble modelling, the focus was on a subset of variables of XSS scripts that can be used as payloads in real-world attacks as well. Data samples are also taken into consideration on the basis of which detection has been performed.

3.2 STAGE TWO – DATA PREPROCESSING:

In the next stage of this cycle, Data cleaning and pre-processing have been carried out. This process has included the removal of noise from data and removal of outliers that were present

in the different datasets. Collected all essential information to model and account for noise. Methodologies for taking care of missing data fields have also been used to balance the datasets. The feature extraction process has been performed based on various parameters like Special symbols or keyword, redirection links, length of the script, sensitive words, etc.

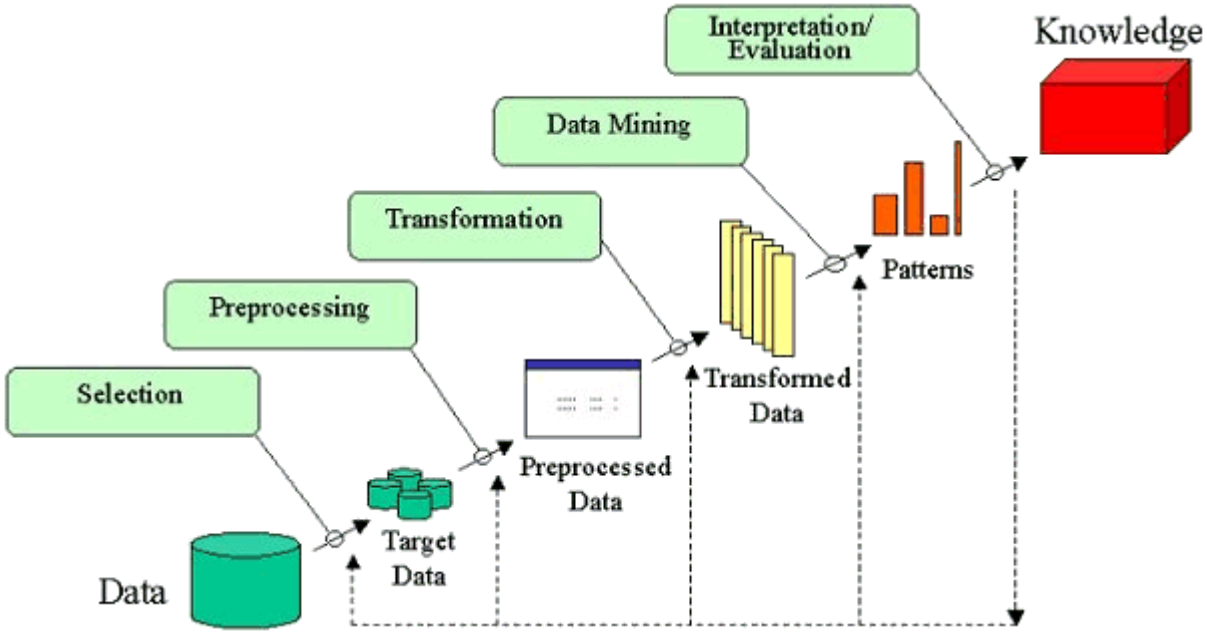

Figure 4: Knowledge Discovery in Databases (KDD) methodology [18].

3.3 STAGE THREE – DATA TRANSFORMATION:

This phase has been focused on data reduction as well as projection. this has achieved by finding useful features to represent the data that has the objective of finding malicious XSS scripts from benign scripts. Along with that, transformation methods have been used to discover invariant representations for the data. The columns have been created which have outputs of different feature extraction functions such as the presence of special keywords and symbols in any data sample. lastly, the datasets have been split into train and test subsets for the application of the model.

3.4 STAGE FOUR – DATA MINING:

The data mining phase has been started with the process of choosing the actual data mining task i.e. extraction of malicious scripts. Hence the goal of the KDD process has already been decided as this was clearly a process of classification. While deciding the data mining algorithm and applying Machine Learning Model various previous works have been taken into consideration. Various researchers have selected different selecting methods for searching patterns in the data. Considering all that, the utilization of the Random Forest Bagging model with various parameters has been decided. This model has been fitting

appropriately with the requirements. Finally, the process of data mining has been performed by applying this model on the dataset with the purpose of searching for patterns of malicious XSS scripts in a particular representational form as per decided classification rules.

3.5 STAGE FIVE – INTERPRETATION AND EVALUATION:

In the last stage of this process, Firstly, this model has been compared with the different Machine Learning models such as Gradient Boosting and Decision Tree Bagging algorithms, and the Interpretation of mined patterns has been carried out. Secondly, the model developed in this research has also been applied on to an imbalanced dataset and compared the results with the balanced ensembled data set created and used in this research. Lastly, the consolidated graph of discovered knowledge has been plotted. Visualization of Accuracy and Training Time of Models has been carried out in this stage.

Design specification, Implementation, and Evaluation phases of this research have illustrated in more detail in the subsequent sections of this report.

4 Design Specification

In this section, the architectural design along with the overall process of the progression for this research has been explained. The designing process of this research has been begun with the searching of the required data from the various sources and merging that into a targeted balanced dataset using ensemble dataset modelling technics. While selecting data, the analysis and the important highlights from past research works have also been considered. Various feature extraction technics have been applied to that data such as binary feature extraction etc. with the intent of detection of the malicious XSS scripts and their signatures. Then the dataset has been split into training and testing subsets with maintaining the ideal ratio of 70:30. Additionally, use of a Random Forest Bagging algorithm has been used which eventually helps to achieve improved stability and accuracy as well as reduced variance. Finally, the performance of the model has been tested and analysed using the test data.

4.1 BLOCK DIAGRAM:

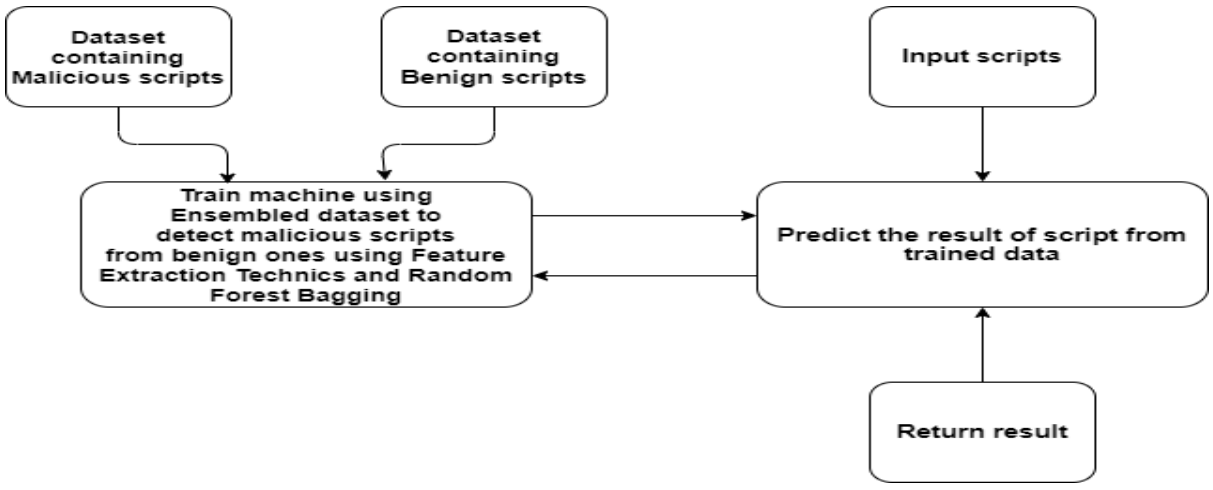

Figure 5: Block Diagram of basic Machine Learning approach used.

4.2 ARCHITECTURAL DIAGRAM / MECHANISM:

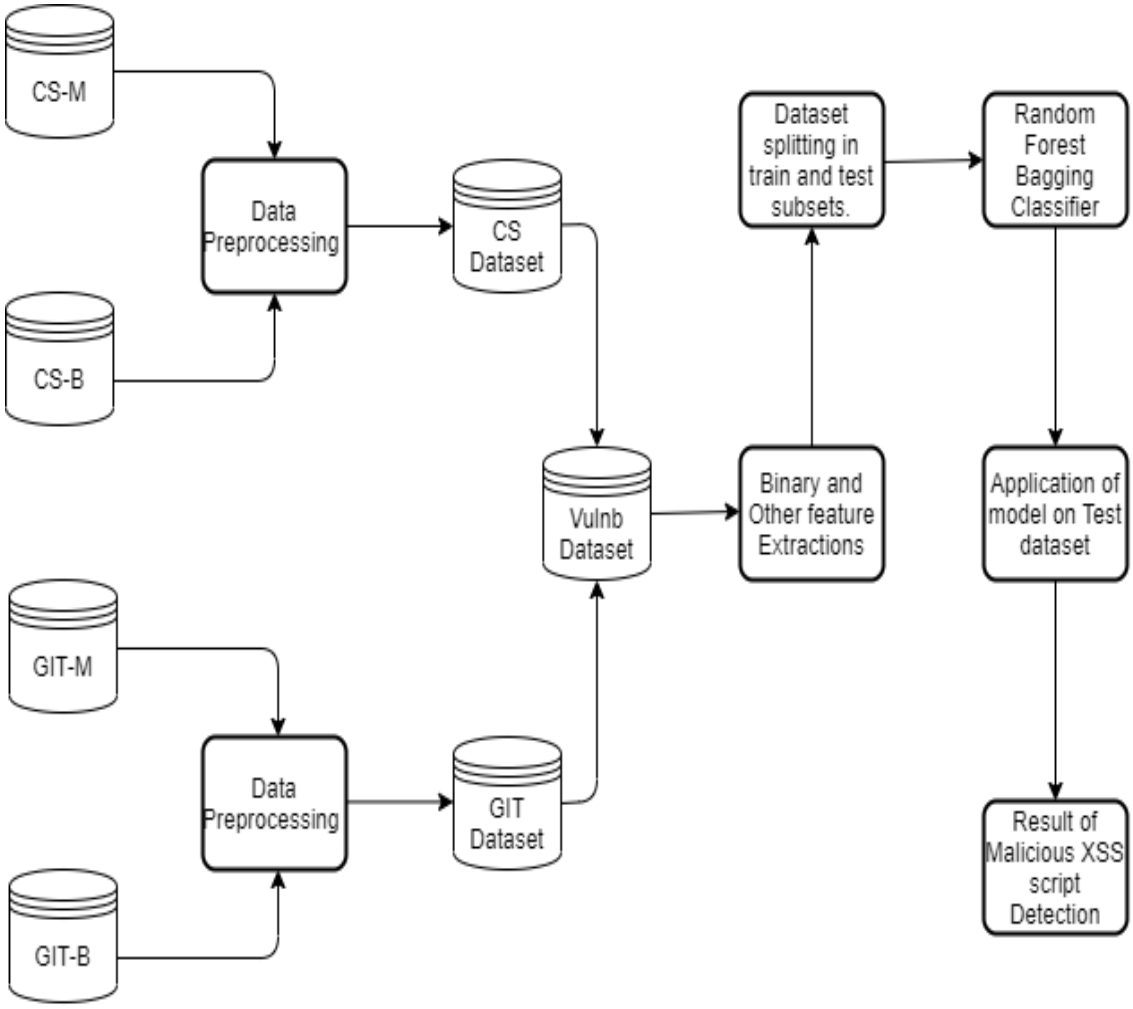

Figure 6: Architecture Diagram of Data Merging, Data Preprocessing along with Extraction of Features & Random Forest Bagging Model.

5 Implementation

This section gives more insights into the actual implementation of this research work. This model for the detection of malicious XSS scripts from the benign inputs has included multiple levels of implementation and development stages have been executed sequentially. This process has been started with the ensemble of the targeted dataset from various sources for achieving an expected outcome using Machine Learning approach. Many data sources have been examined and some of them are used in the targeted dataset of this project such as Kaggle open dataset and GitHub open dataset repositories of XSS cheat sheets. These added payloads and cheat sheets data help a lot to detect real-time XSS attacks. These processes have been done by using the manual methods. Using the manual approach duplicated data has been removed into the excel and then converted into the CSV and TXT format for further utilization. The dataset and the sources that have been used for the targeted dataset are the latest and contain a wide range of XSS scripts and payloads. While collecting the data from various data sources, the data has been accumulated from many files as well as different formats. Hence the process of combining all that data into a targeted dataset has been carried out using the Python scripts. Jupyter-Notebook, the web-based computing environment has been used for the coding and running along with describing Data analysis. Python 3.8.5 has been used for the scripting. The major reasons for selecting this language were that it is a stable language having simple syntaxes which makes it easy to learn. Python encompasses a huge community that giving a continuous contribution to it. Hence various libraries of python are readily accessible on the internet including the libraries for machine learning as well.

In the next stage of this process, the Data has been read from the text files like TXT's and CSV's to store in a data frame. The data frame has been loaded from CSV's with the help of the panda's package of python which provides the facility to read data from CSV. Then the format of data has been changed from a list of dictionaries to a data frame. Then the focus of implementation was on the creation of Functions to extract various features from the Scripts Such as the length of the script, check if redirection link is present. Moreover, checks for the presence of special keywords and special symbols have been carried out. Finally, a number of sensitive keywords including alert, script, onerror, confirm, img, onload, eval, prompt, src, href, javascript, window, fromcharcode, document, onmouseover, cookie, domain, onfocus, expression, iframe, onclick, %3c, %3e, and sensitive characters including [<>,\'\"/] has been recorded. These all special keywords and special symbols have been shortlisted with the reference of previous studies which have been discussed in the Related Works section of this report. Then using these functions, the features have been extracted from the scripts for application of Machine Learning. The python machine learning package ‘sklearn’ has been used which provides various classifier models and algorithms as well as multiple functionalities such as train_test_split and KFold which has been utilized in this study to split the data into training and test datasets. Subsequently, this ensemble and preprocessed data have been given as input for the Random Forest model using one of the sklearn-libraries.

```
In [1]:
import numpy as np
import pandas as pd

import re
import time
import random

from sklearn.utils import shuffle
from sklearn.model_selection import train_test_split, cross_val_score, KFold
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from sklearn.ensemble import RandomForestClassifier, BaggingClassifier, GradientBoostingClassifier
from sklearn.tree import DecisionTreeClassifier

import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

Figure 7: Various packages and libraries used in this implementation.

Random Forest model applies bagging technics of ML for processing on the given dataset and predicts the outputs. The outcomes of the model get generated in the form of metrics using the libraries of the same sklearn package like accuracy_score, confusion_matrix, classification_report. Lastly, the matplotlib library along with the seaborn library have been used for carrying out the tests of correlation. The seaborn lib is utilized to generate the Graph of the comparison of the results carried out from various models like Decision tree Bagging and gradient boosting models with Random forest. This comparison graphs clearly advocates the use of the Random Forest model in this research that this model always gives better Accuracy as compared to other models.

6 Evaluation

Evaluation of this research work has been carried out by deriving and assessing the confusion matrixes of the models. A confusion matrix is a table that used to visualize the overall performance of any classification model. It can be utilized to calculate Accuracy, Sensitivity (also called as recall), Specificity, and Precision. The following diagram helps to understand the confusion matrix in detail.

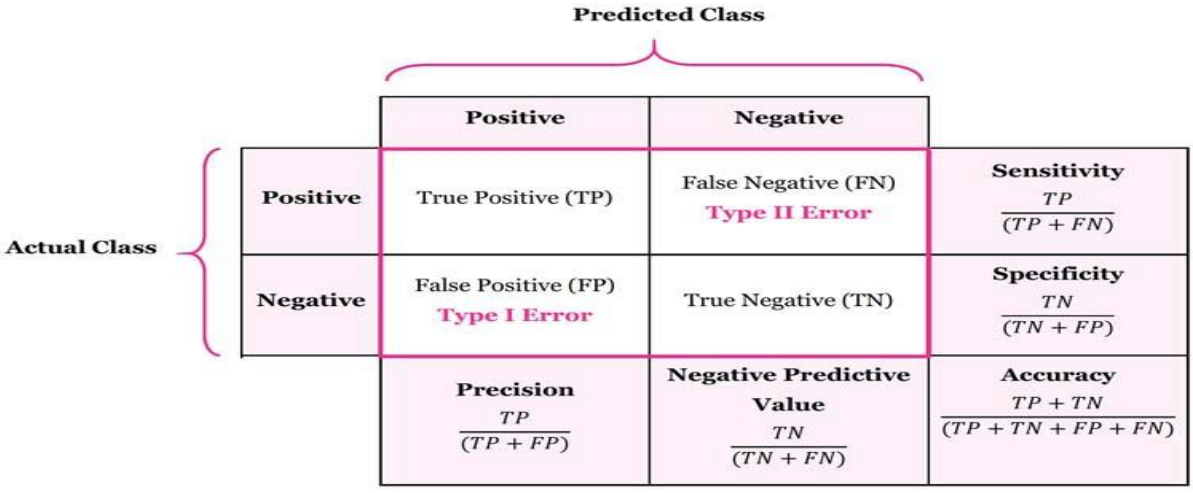

Figure 8: Confusion matrix that uses for the evaluation [21].

The metrics for this study of detection of the Malicious XSS scripts are as follows:

• TP: Malicious XSS scripts and model predicted the malicious XSS script.
• TN: Benign script and model predicted the non-malicious / benign script.
• FP: Benign script and model predicted the malicious XSS script.
• FN: Malicious XSS script and model predicted the non-malicious / benign script.

Hence the calculated output using stated metrics for this model is as follows:

Balanced and Ensemble Dataset	***Accuracy***	***Precision***	***Recall***
No	99.48%	100%	100%
Yes	97.16%	96%	97%

Table 1: Outcomes with and without balanced and ensembled dataset.

Algorithm used	***Accuracy***	***Precision***	***Recall***
Random Forests	97.16%	96%	97%
Decision Tree + Bagging	96.65%	96%	97%
Gradient Boosting	95.74%	94%	96%
Decision Tree	95.43%	95%	97%

Table 2: Accuracy of the different models as compare to Random Forest.

Following are the Three experimental case studies have been taken into consideration while evaluation of this model:

6.1 Case Study 1: Performance while using balanced and unbalanced data

The created model has been applied to the original dataset which was totally imbalanced as the number of malicious XSS scripts and non-malicious XSS scripts has a big difference. Table 1 gives a clear idea that the accuracy is very high in case of imbalanced data. the accuracy, recall and precision are appearing approximately 100% which shows that the model has fizzled to detect the malicious XSS scripts. Because of the highly imbalanced dataset, the model has become one-sided and became overfitted. To manage this problem the data has been balanced and then assessed. After evaluating the model with this balanced ensemble dataset, the outcomes are improved as expected. The accuracy has been recorded up to 97.16% where precision and recall have been recorded up to 96% and 98% respectively. These outcomes are way better than the imbalanced dataset. It can be concluded from these outcomes that the balanced and ensemble data always gives a better result compare to imbalanced data while using same Random Forest algorithm for both.

6.2 Case Study 2: Performance compared to other Algorithms

It can be clearly observed from table 2 that using the same balanced data on various algorithms and machine learning technics, Random Forest performs well in terms of Accuracy, Precision, and Recall. The following graph has been plotted using the seaborn lib that gives a clear idea about the comparison of the results carried out from various models like Decision tree Bagging and gradient boosting models with Random forest.

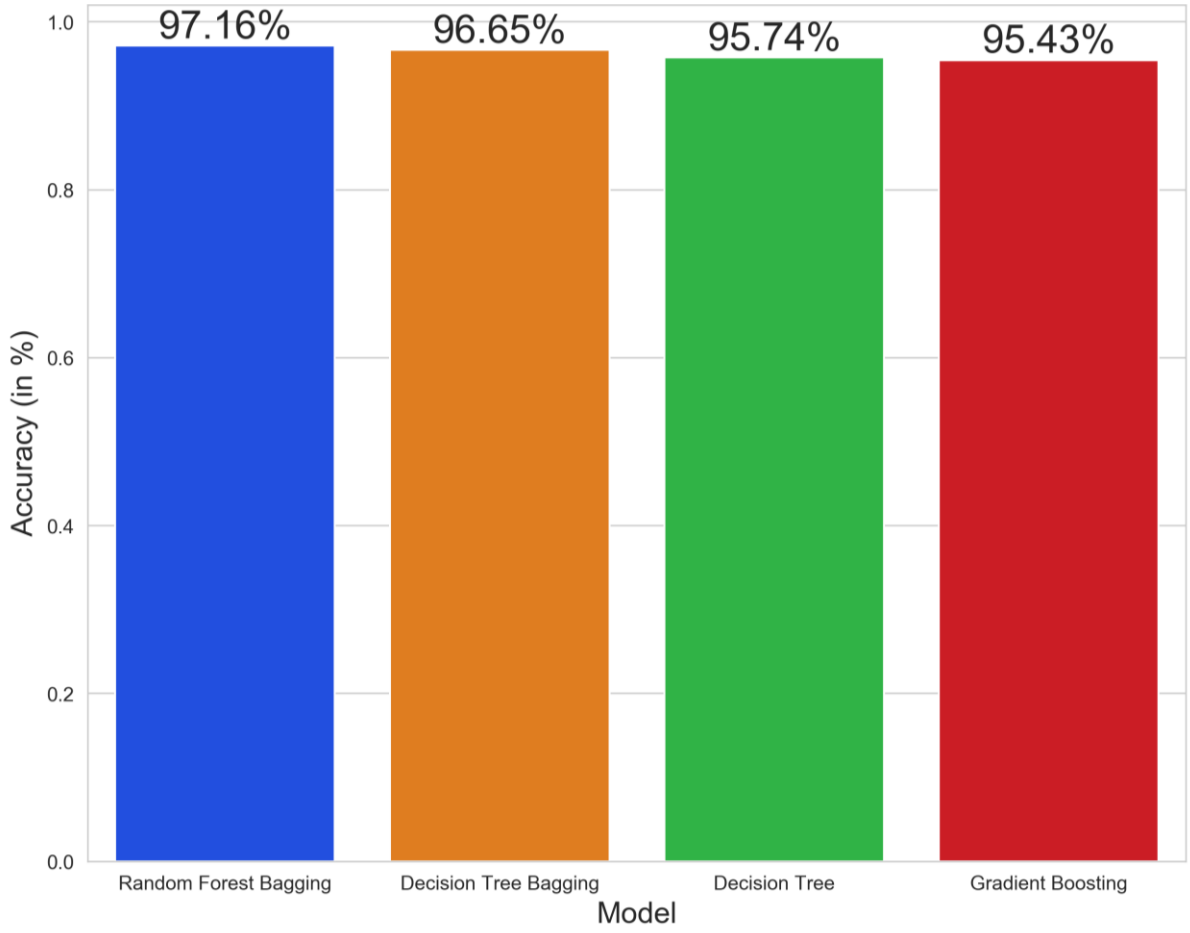

Figure 9: Comparison of Accuracies.

6.3 Case Study 3: Training time required compared to other Algorithms

The training time for each model has also been recorded and the following graph has been plotted. From the graph, it can observe that the Random Forest model is required more time to get trained. However, it should notice and understand that the timings given in the graph are in seconds, so the difference is not that big. Secondly. The system configuration on which the machine is getting trained also makes a visible impact on this prediction so the difference between the results on the high-end server will be negligible in this case.

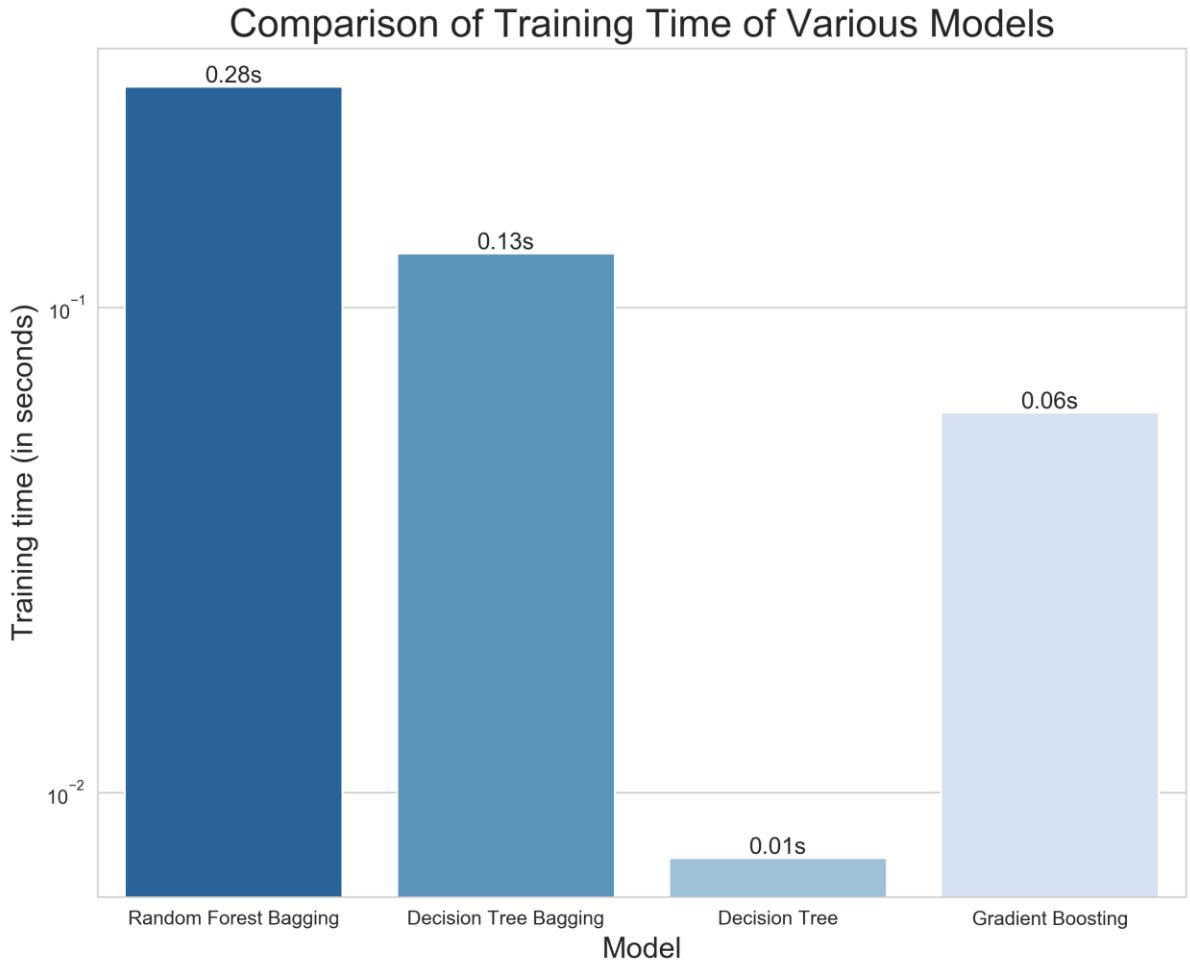

Figure 10: Comparison of Training-Times of different models.

6.4 Discussion:

It can be clearly observed that because of the utilization of a balanced and ensembled dataset, there was a nice enhancement has been recorded in the overall confusion matrix. Balancing and ensemble methods make a difference to manage the restrictions like limited data availability which eventually helps to enhance the capability of the prediction of this model.
The evaluation results of this research show that there is no such ideal machine learning approach for such problems. However, the bagging approach of the Random Forest model has been chosen as it gives higher accuracy as compared to others and provides consistent results. the ensemble dataset also complements the machine learning approach and improves the overall outcomes of this model of detection of the malicious XSS scripts.

7 Conclusion and Future Work

In this paper, an XSS attack detection method based on an ensemble dataset modelling approach has been developed. The balanced ensemble dataset has been created for this work utilizing multiple data sources of XSS payloads along with the XSS datasets. To simulate real attack scenarios, different percentages of sampled malicious XSS script records has been inserted in the dataset. Knowledge has been acquired from ontology to abstract features of the XSS with the help of earlier research to extract features and has a satisfactory results which can be useful in the Information Security domain up to some level for the detection of the malicious XSS scripts. To increase the generalization of this study, the bagging method using the Random forest algorithm has been utilized to get more stability, accuracy, and reduces variance which also helps to avoid overfitting of the model. To further explain the detection results, the model developed in this research has also been applied to an imbalanced dataset and compared the results with the balanced ensembled data which clearly showed the advantages of Balanced and Ensemble data in terms of better outcomes. Moreover, this model has been compared with the different Machine Learning models such as Gradient Boosting, Decision Tree Bagging algorithms and the consolidated graph of discovered knowledge has been plotted which clearly advocates the use of Random Forest over the other models. The results showed this method performed well as compared to the other methods in most of the cases and achieved 97.16% accuracy even in the worst case and even after using a balanced ensembled dataset.

In future work, this method can get developed using a Deep learning approach. Moreover, this is a detection approach, hence this method can get integrated within real-world security risk prevention systems such as intrusion prevention systems (IPS).

Acknowledgment

I would like to use this moment to thank my project guide and supervisor Niall Heffernan for every insights and recommendation that have brought more value to this work. the encouragement is given to me for conducting this study and helping me to concentrate on the main element of this study really helped me a lot. I feel thankful because of his commendable mentorship, which truly played a major role in this overall journey. I also would like to express my gratitude towards our course director Dr. Arghir Moldovan for helping in every crucial situation with his valuable expertise and excellent encouragement throughout this MSc course at the National College of Ireland.

References

[1] M. Dayal Ambedkar, N. S. Ambedkar and R. S. Raw, "A comprehensive inspection of cross site scripting attack," 2016 International Conference on Computing, Communication and Automation (ICCCA), Noida, 2016, pp. 497-502.

[2] J. Williams and D. Wichers, "OWASP top 10–2013," OWASP Foundation, 2013

[3] X. Guo, S. Jin and Y. Zhang, "XSS Vulnerability Detection Using Optimized Attack Vector Repertory," 2015 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, Xi'an, 2015, pp. 29-36, doi: 10.1109/CyberC.2015.50.

[4] S. Akaishi and R. Uda, "Classification of XSS Attacks by Machine Learning with Frequency of Appearance and Co-occurrence," 2019 53rd Annual Conference on Information Sciences and Systems (CISS), Baltimore, MD, USA, 2019, pp. 1-6, doi: 10.1109/CISS.2019.8693047.

[5] Yun Zhou and Peichao Wang, "An ensemble learning approach for XSS attack detection with domain knowledge and threat intelligence," 2019 ScienceDirect, Computer and Security, volume 82, May 2019, pages 261-269.

[6] A. E. Nunan, E. Souto, E. M. dos Santos and E. Feitosa, "Automatic Classification of Cross-Site Scripting in Web Pages Using Document-based and URL-based Features", IEEE Symposium on Computers and Communications (ISCC), pp. 702-707, 2012.

[7] H. He and E. A. Garcia, "Learning from Imbalanced Data", IEEE Transactions on Knowledge and Data Engineering, vol. 21, no. 9, pp. 1263-1284, 2009.

[8] N. V. Chawla, K. W. Bowyer, L. O. Hall and W. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique", Journal of Artificial Intelligence Research, 16, 321-357, 2002. (Pubitemid 43057176)

[9] Y. Liu, M. Shriberg, "Comparing Evaluation Metrics for Sentence Boundary Detection",IEEE In Proceeding International Conference On Acoustic, Speech And Signal Processing, 2007.

[10] T. Kaiho, T. Matsuda, M. Sonoda and J. Chao, "Feature Extraction of Embedded URL Cross-Site Scripting Attacks", The 77th National Convention of IPSJ, vol. 1, pp. 427-428, 2015.

[11] T. Matsuda, D. Koizumi and M. Sonoda, "Cross Site Scripting Attacks Detection Algorithm Based on the Appearance Position of Characters", The 5th International Conference on Communications Computers and Applications (MIC-CCA2012), pp. 65-70, 2012.

[12] Byung-Ik Kim, Chae-Tae Im and Hyun-Chul Jung, "Suspicious malicious web site detection with strength analysis of a javascript obfuscation", International Journal of Advanced Science & Technology, vol. 26, pp. 19-32, 2011.

[13] Peter Likarish, Eunjin Jung and Insoon Jo, "Obfuscated malicious javascript detection using classification techniques", Proceedings of the 4th International Conference on Malicious and Unwanted Software (MALWARE), pp. 47-54, 2009.

[14] A. Fazelpour, T. M. Khoshgoftaar, D. J. Dittman and A. Napolitano, "Investigating New Bootstrapping Approaches of Bagging Classifiers to Account for Class Imbalance in Bioinformatics Datasets," 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA), Miami, FL, 2015, pp. 987-994.

[15] S. Nagi, D. K. Bhattacharyya, "Classification of microarray cancer data using ensemble approach", Network Modeling Analysis in Health Informatics and Bioinformatics, vol. 2, no. 3, pp. 159-173, 2013.

[16] A. C. Tan, D. Gilbert, "Ensemble machine learning on gene expression data for cancer classification", 2003.

[17] C. Chen, A. Liaw, L. Breiman, "Using random forest to leran imbalanced data", Department of Statistics University of California - Berkeley East Lansing Michigan Tech. Rep., 2006.

[18] Fayyad, Piatetsky-Shapiro, Smyth, "From Data Mining to Knowledge Discovery: An Overview", in Fayyad, Piatetsky-Shapiro, Smyth, Uthurusamy, Advances in Knowledge Discovery and Data Mining, AAAI Press / The MIT Press, Menlo Park, CA, 1996, pp.1-34

[19] Syed Saqlain Hussain Shah, 17 March 2020. [Online]. Available: https://www.kaggle.com/syedsaqlainhussain/cross-site-scripting-xss-dataset-for-deep-learning/data.

[20] Duoergun,, 10 Sept 2017. [Online]. Available: https://github.com/duoergun0729/1book/tree/master/data.

[21] Abhigyan, 17 Mar 2017. [Online]. Available: https://medium.com/analytics-vidhya/calculating-accuracy-of-an-ml-model-8ae7894802e

[22] Rathore, Shailendra et al. “XSSClassifier: An Efficient XSS Attack Detection Approach Based on Machine Learning Classifier on SNSs.” *J. Inf. Process. Syst.* 13 (2017): 1014-1028.