IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 8.102 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 6, June 2023

DOI: 10.17148/IJARCCE.2023.126119

AI to Predict Phishing Attacks on Edge Devices

Akashdeep Boxi[1], Lakhan Kumar[2], Rishi Singh[3]

BE, Department of CSE, Atria Institute of Technology, Bangalore, India [1]

BE, Department of CSE, SVIT, Bangalore, India [2]

BE, Department of Mechanical, PESU, Bangalore, India [3]

Abstract The world of today is a digital one, where decisions are made online and the internet plays a significant role in both society and the economy. Naturally, the security of the internet, or more specifically, the security of the World Wide Web (www), is crucial and revolutionary. The internet is frequently open to attacks from potential hackers who attempt to infiltrate the system in question in order to illegally take advantage of the system's resources. These attacks, sometimes known as web attacks, are well-known in the computer community and are a major issue. Though there are several existing systems to counter the problem of attacks on the web, most of these systems have their own drawbacks, as in they do not provide classification on any other grounds except frequency, thus causing many web attacking http requests to fall out of the bracket. Our project's goal is to identify these web attacks from http requests based on a variety of characteristics and determine whether they qualify as web attacks or not. In order to add originality, we also intend to further categorise the attacks as HTML, JavaScript, or SQL attacks. Thus, the system resolves the issue of hidden web attacks via http requests, greatly enhancing the system's security.

Keywords: Supply World Wide Web (www), web attacks, web attacking http requests, HTML, JavaScript or SQL attacks.

I. INTRODUCTION

With the use of internet technology, the need of traditional methods to complete various jobs has decreased. The use of web apps has also increased dramatically and is found in many different industries worldwide. The use of cloud-based systems has made it simpler to maintain apps. Each web application, whether it is hosted in the cloud or on a dedicated server, will have its own unique URL. The likelihood of the system being attacked and a server being compromised is higher. Web server attacks are a common type of attack, accounting for 75% of all attacks. There are two distinct paradigms of advantages from web attacks, the first of which is access to end users' personal information, and the second of which is injecting script that can hack data when users click or carry out certain downloads. Every web application created by different businesses must use protocols to secure the system. However, by recognising and categorising them by reading the request's properties and then carrying out the machine learning method, the prevalence of such assaults can be reduced.[1]

II. OBJECTIVES

i. To accomplish the classification of HTTP request into anomaly or non-Anomaly and then anomaly is classified into HTML, SQL and JavaScript classifier.

ii. To minimize the amount of security issues for a web application.

iii. To implement a software web application which will have screens to view the data sets, stop words, addition or removal of stop words, and create a view for clean data sets, tokenization, and frequency. [2]

iv. Design and implementation of weight feature vector computation in which the clean data sets are converted into sequence of words followed by computation of frequency, and then IDFT and FV are found.

III. PROBLEM STATEMENT AND PROPOSED SOLUTION

"To detect weather the HTTP request is malicious or not, using request attributes and classify it as a SQL, HTML or JavaScript anomaly by using Machine learning". Our aim is to detect whether the HTTP request is malicious or not, where word vector along with neural network is used with the hidden layers along with TF-IDF methodology. The word vector block and TF-IDF are combined together in order to determine whether a given request is a web attack or it is not a web attack. There are different categories of clusters which are considered in order to perform the category checks for the web attacks and the category includes the SQL, HTML and JavaScript.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 8.102 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 6, June 2023

DOI: 10.17148/IJARCCE.2023.126119

IV. SYSTEM ARCHITECTURE

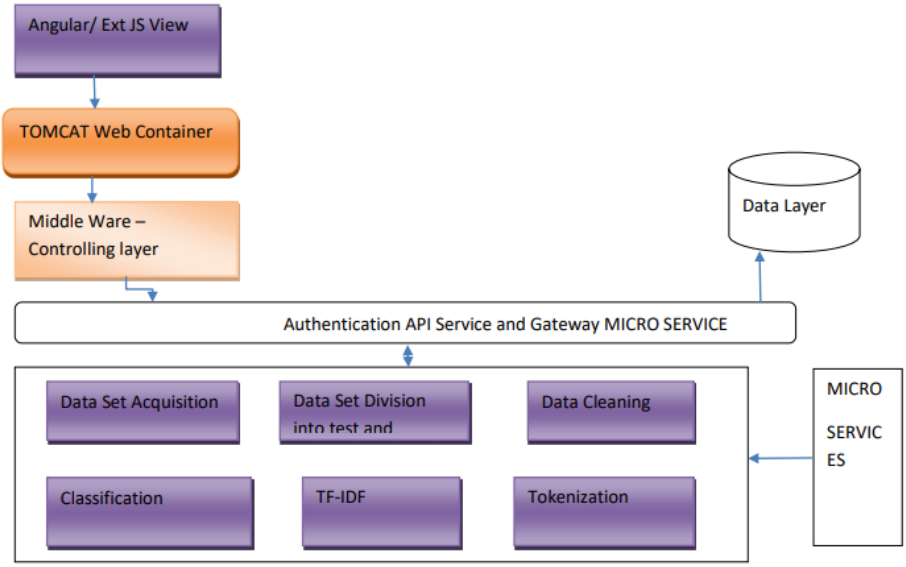

Figure 1 shows the system architecture of the proposed system and the following layers.

Figure 1 shows the system architecture. As shown in the figure there are various components which are involved in the working model for the project. These components are as follows:
Angular/Ext JS View For the development of the front end us done with the help of using angular and Ext JS framework along with java server pages.
TOMCAT Web Container: There are many servers available in the market which is responsible for handling the web requests. Most of the other servers are heavy weight and also are commercial in nature. Here we make use of open source and light weight tomcat server.
Middle Ware – Controlling layer: The request parameters from the front end and URL will be validated. If the request URL is valid then the form data will be validated and then, the request forwarding is done to the authentication layer. This also performs the basic validations like empty checks and regex validations. If any validation fails, then response is sent to the front end otherwise the request is forwarded to the authentication layer and respective services. [2-5]
Data Layer: The data layer is responsible for storage of information related to registered users, admin, doctors. The data layer will also be able to store answers of users, appointment information as well as the classification information.
Authentication Layer: The users request is validated and after it is being checked whether the request contains valid application id and also has a valid session. User will be thrown out if the session is invalid.
Micro services: Each of the micro services are independent executing the business logic for the specific algorithm which can be data set collection operations, data cleaning operations, word stream operations, word stream count, inverse word stream count.

V. SYSTEM DESIGN

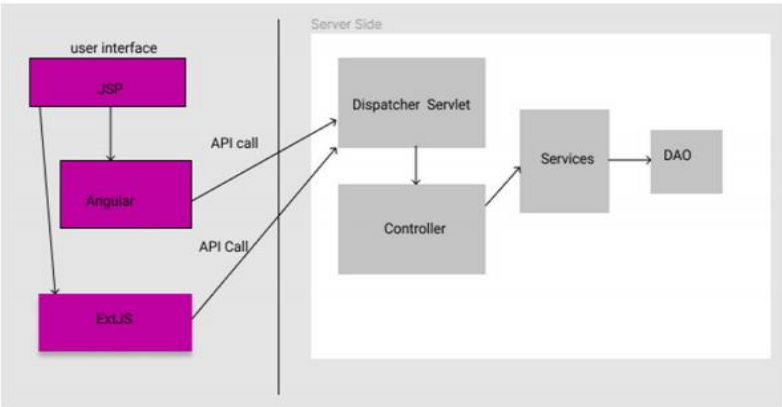

Figure 2 shows the UI of the project.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 8.102 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 6, June 2023

DOI: 10.17148/IJARCCE.2023.126119

A UI is the key for the user to interact with the system. Figure 2 shows the interface architecture. As shown in the figure JSP page will refer angular as well as Ext JS frameworks to design the user interface and then API call will be made which will first go to dispatcher servlet which does high level validations, followed by controller which will have all the basic and advance validations, services will have all algorithms and business logic and DAO responsible for CRUD (Create Retrieve Update and Delete) operations. [6]

The algorithm responsible for storing undesired phrases in the program is shown in Figure 3. Creating Unwanted Words algorithm is in charge of coming up with an unwelcome word that isn't in the list of conventional unwanted words. If the unwanted term already exists, a validation error will be displayed; otherwise, the unwanted word will be produced.

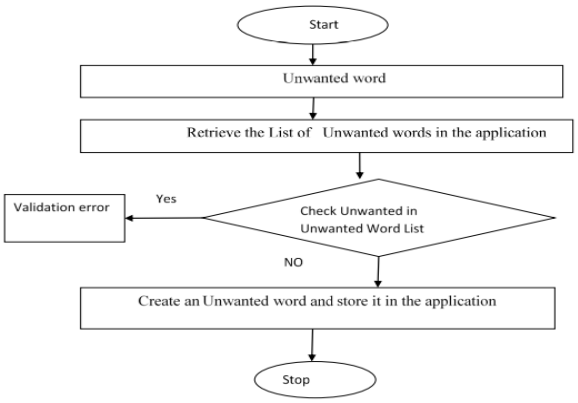

Figure 3 shows the algorithm responsible for storing undesired phrases in the program.

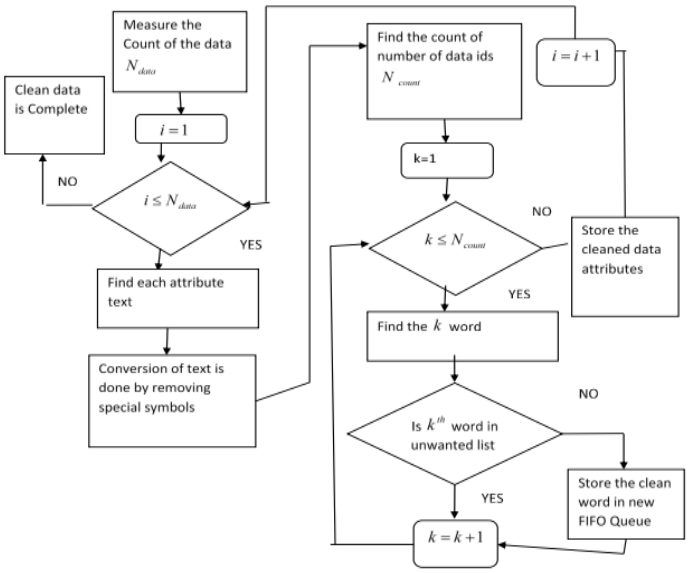

Figure 4 depicts the quantity of data rows counted.

The major job is to employ clean data, which is obtained by cleaning each attribute by deleting any unneeded words and then removing any residual special symbols. Figure 4 depicts the quantity of data rows counted. From the first word to the last number of words, the process is repeated. After that, special symbols are deleted from each text attribute, and the number of counts for data ids is calculated. From the beginning of each word to the number of words in FIFO, each word is taken, checked, and shown to see whether there are any undesired words in the list. If the word is present, it is skipped; otherwise, it is added to the queue and incremented. This process continues till the task is accomplished.[7]

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 8.102 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 6, June 2023

DOI: 10.17148/IJARCCE.2023.126119

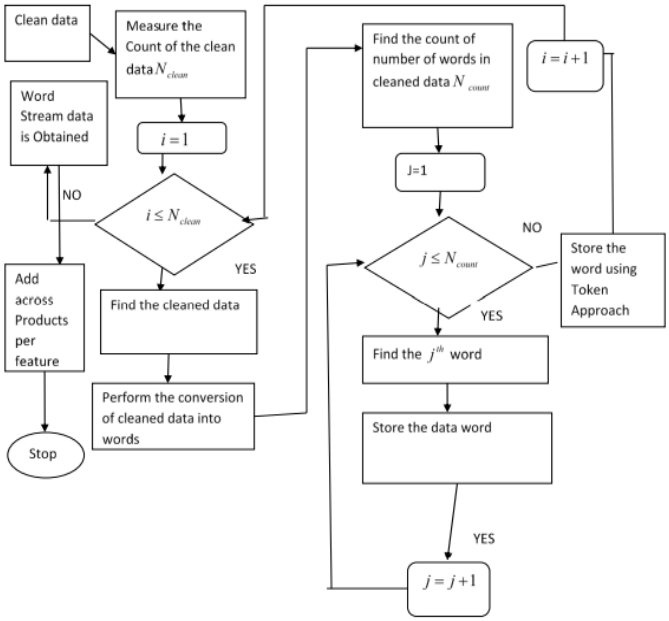

Figure 5 depicts word processing.

The conversion of cleaned text into a series of word stream values is handled by this streaming procedure. Figure 5 depicts word processing, in which cleaned text is obtained first, followed by conversion in a FIFO queue using a Delimiter. By that time, we've figured out how many words we've gotten. Now we classify each word, starting with the first and ending with the last, and save it with a unique representation id. Cleansed text is converted into a set of word stream values, which are then stored as a numeric value to represent count. The word stream processing is shown in Figure 6. First, a list of unique data ids is compiled, and the corresponding count is determined. Each unique id is found starting with the first word and ending with the last word. Now we get the number of count of each unique word from the first data id list; this count of words repetition is done on specific data ids, and each word stream count is saved.

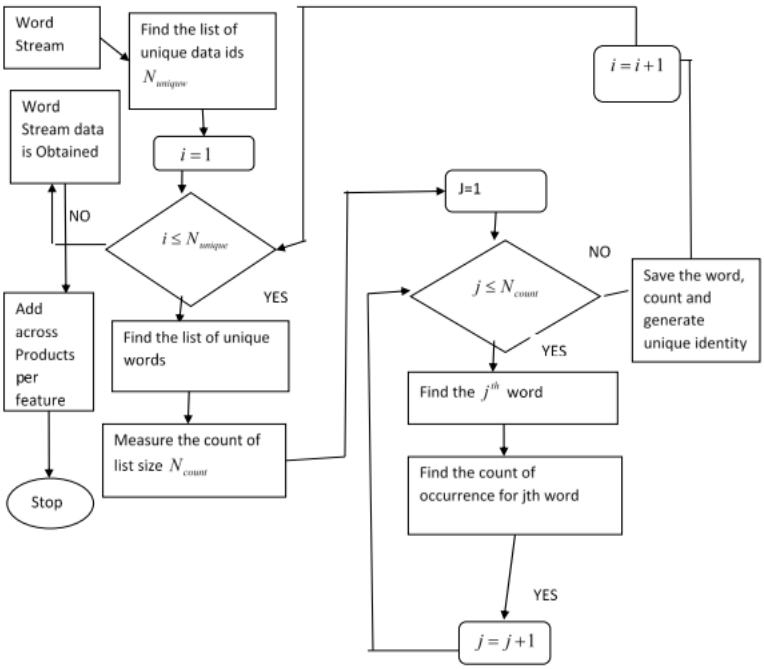

Figure 6 represents word stream count processing.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 8.102 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 6, June 2023

DOI: 10.17148/IJARCCE.2023.126119

Algorithm: K-means Clustering:

- Purpose: Classification of HTTP request into anomaly or not.
- Input: The classified words related to various categories along with the total FV score for each category specific to dataset.
- Output: HTTP request is classified as anomaly and non-anomaly, and then based on the total FV score, anomaly is further classified into html, JavaScript and SQL.
- Different words related to various kinds of web attacks are obtained from the training vectors.
- Unique data ids are found by making use of TF-IDF matrix.
- The count of various web attack category words is obtained.
- The total feature vector for each of the data set based on the word and TF-IDF values is found.
- Compute the distance between the feature vector and the trained vectors.
- Find the minimum distance.
- The class label corresponding to the minimum distance is assigned a class respectively

Data Structure Design:

● JSON

JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. It is language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Python, and many others.

● JSON Web Tokens (JWT).

JSON Web Token (JWT) is an open standard (RFC 7519) that defines a compact and self-contained way for securely transmitting information between parties as a JSON object. This information can be verified and trusted because it is digitally signed. JWTs can be signed using a secret (with the HMAC algorithm) or a public/private key pair using RSA or ECDSA.

VI. RESULTS

The deep learning model using K-means clustering was used to build a web-attack detection system. Figure 7 shows the classification output of the project. Figure 8 shows the accuracy of the model after complete execution. To accomplish the classification of HTTP request into anomaly or non-Anomaly and then anomaly is classified into HTML, SQL and JavaScript classifier.

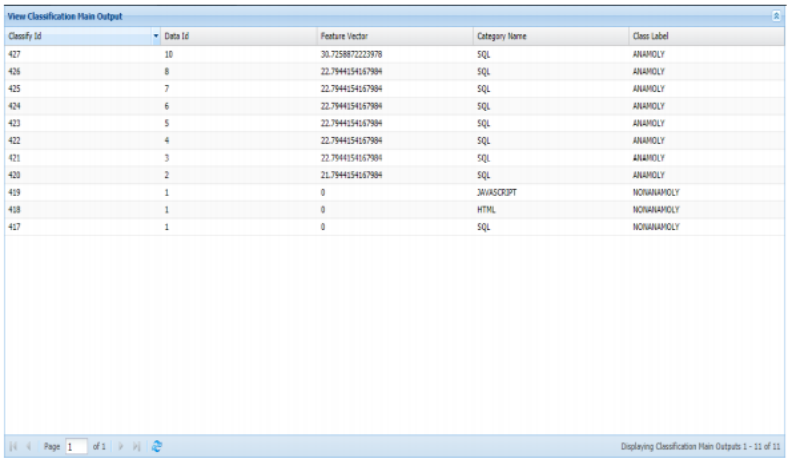

Figure 7 shows the classification output of the project.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 8.102 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 6, June 2023

DOI: 10.17148/IJARCCE.2023.126119

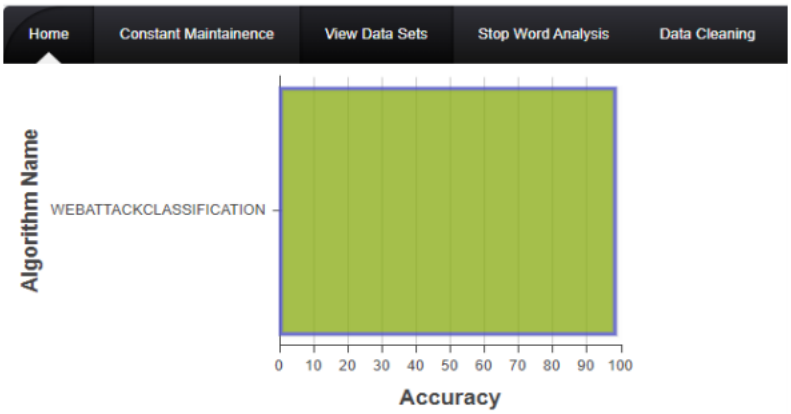

Figure 8 shows the accuracy measure of the model after execution.

VII. CONCLUSIONS

Our system will take the data sets which contains web-attack and non-web-attack. The data sets undergo cleaning for removal of unwanted words and special symbols takes. After that the clean description is converted into a sequence of words. Then, redundancy is removed by doing a word stream count computation. After that weighted stream and number of datasets in which the word is present is calculated. Then, classification of data is done by using K-means algorithm. We have detected web attacks from the http requests based on many parameters, and classify them as web attacks or not. We also have classified the attacks as HTML, JavaScript or SQL attacks, thus providing a novelty. Thus, the system solves the problem of undetected web attacks through http requests and thus increases the security of the system.

VIII. FUTURE SCOPE

The main disadvantage of this project is that it was completed using a limited amount of datasets and no custom datasets were used. This project could be improved even further by including a subjective category for web attacks. Further tune the model for better performance.

REFERENCES

[1] R. Munadi, T. S. Fajri, E. D. Meutia and E. Mustafa, "Analysis of SQL injection attack in web service (a case study of website in Aceh province)," 2013 3rd International Conference on Instrumentation, Communications, Information Technology and Biomedical Engineering (ICICIBME), Bandung, 2013, pp. 431-435, doi: 10.1109/ICICI-BME.2013.6698541.

[2] K. R. Kishore, M. Mallesh, G. Jyostna, P. R. L. Eswari and S. S. Sarma, "Browser JS Guard: Detects and defends against Malicious JavaScript injection based drive by download attacks," The Fifth International Conference on the Applications of Digital Information and Web Technologies (ICADIWT 2014), Bangalore, 2014, pp. 92-100, doi: 10.1109/ICADIWT.2014.6814705.

[3] M. Y. Kim, and D. H. Lee. (2014). Data-mining based SQL injection attack detection using internal query trees. Expert Systems with Applications. 41(11), pp: 5416-5430.

[4] Yuan, Zhenlong & Lu, Yongqiang & Wang, Zhaoguo & Xue, Yibo. (2014). Droid-Sec: Deep Learning in Android Malware Detection. ACM SIGCOMM Computer Communication Review. 10.1145/2619239.2631434.

[5] J. Saxe, and K. Berlin, "Deep neural network based malware detection using two dimensional binary program features," in International Conference on Malicious and Unwanted Software (MALWARE), IEEE, 2015.

[6] R. M. Pandurang and D. C. Karia, "A mapping-based podel for preventing Cross site scripting and sql injection attacks on web application and its impact analysis," 2015 1st International Conference on Next Generation Computing Technologies (NGCT), Dehradun, 2015, pp. 414-418, doi: 10.1109/NGCT.2015.7375152.

[7] T. Rashid, I. Agrafiotis, and J. RC Nurse, "A new take on detecting insider threats: exploring the use of hidden markov models," in Proceedings of the 8th ACM CCS International Workshop on Managing Insider Security Threats, ACM, 2016.