# Log-based Intrusion Detection System using Machine Learning

MSc Research Project
Artificial Intelligence

Sonia Francis Javior
Student ID: x22175903

School of Computing
National College of Ireland

Supervisor: Rejwanul Haque

**National College of Ireland**
**Project Submission Sheet**
**School of Computing**

| | |
|---|---|
| **Student Name:** | Sonia Francis Javior |
| **Student ID:** | x22175903 |
| **Programme:** | Artificial Intelligence |
| **Year:** | 2023 |
| **Module:** | MSc Research Project |
| **Supervisor:** | Rejwanul Haque |
| **Submission Due Date:** | 14/12/2023 |
| **Project Title:** | Log-based Intrusion Detection System using Machine Learning |
| **Word Count:** | 7758 |
| **Page Count:** | 18 |

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.

| | |
|---|---|
| **Signature:** | Sonia Francis Javior |
| **Date:** | 31st January 2024 |

## 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): | |

# Log-based Intrusion Detection System using Machine Learning

Sonia Francis Javior
x22175903

## Abstract

The increasing proportion of cyber-attacks has demanded the development of more secure and effective Intrusion Detection Systems (IDSs). The traditional intrusions was more focused on the network layer, but attack penetration was in-depth in the application layer as well. Intrusion detection systems (IDS) are critical when safeguarding the system networks against malicious threats. As the attackers have learned the traditional accessing method the need for developing complex intrusion systems is so necessary. The traditional Intrusion Detection System was more focused on the network layer, but attack penetration was in-depth in the application layer. To solve this gap, an intrusion-based detection system with the help of log files is designed for detecting web attacks. Log file plays a crucial part in this paper since it records the errors and intrusions that happen in the system. This paper discusses the selection of the significant features that are intended to classify the attack. Managing the log files and selecting the significant features in the data are intended to classify the attack. Multiple datasets are generated and it is pre-processed and trained to learn the complex attack patterns of the network system. With the support of the information gained through the generation of logs, the system identifies the vulnerable activities both in the network and application layer. In this research, nine different types of web attacks are detected using supervised machine learning algorithms. The evaluation of these algorithms has been done by considering the precision, recall, F1 -Score and Accuracy factors.

## 1 Introduction

In the modern era of digitalization, every individual is indivisible from the internet. We all are united by technology, which plays a vital role in our day-to-day livelihood. Emerging technologies have made the world interconnected through social media, cloud services, transactions, and other computerized systems Zegzhda et al. (2020). However, all these technological advancements have led to surged cyber crimes and threats. In this era with the increasing number of users becoming part of the digital world, On the Other hand, attackers are also growing along a similar ratio. Cyber criminals have gained the benefit of misaligned networks during the time of pandemic period. In the year 2020, malware attacks rose to 358% [1]in comparison with 2019. Digital attacks are being performed to steal individual privacy or to access the private data of the organization as it can lead to loss of reputation and financial loss. The progression of cyber threats is always

[1] https://aag-it.com/the-latest-cyber-crime-statistics/

associated with the extended progression of technology. So, if a cyber crime happens, there is a medium of technological solution to address it. Therefore, it is critical to safeguard the integrity, confidentiality, and availability of valuable data when defending against cyber-attacks. Several comprehensive approaches and general prevention methods have been implemented within the organization such as preserving the efficiency of the Inventory Control System Devices, partitioning the networks into segments and adding security firewalls to them, implementing secure access methods, creating role-based access controls, and most importantly keeping an eye into updating system log filesBendovschi (2015). However, manifold difficulties arise when it comes to secure web-based attacks, particularly in the application layer of the network.

Setting up the web application firewall has a few limitations and failures which may cause the slow-down of the working application because resources are limited due to the processing of incoming traffic. In addition to that, the web application firewall needs to be regularly maintained and the processing system must be updated to get away from emerging threats and not keeping up the security policies can open to attacking areas. An alternative way of preventing brutal web attacks in the application layer is (Intrusion Detection Systems (IDS) which have recently gained a considerable amount of interest and have played a crucial role in improving web application security. Log files play an important role in discovering and making it less rigorous to web attacks. Logs captured on the web server can help to stop the threats and can identify the user system details. Also, these files are required for identifying the causes of the potential effects that can happen in the future. Analyzing log files can provide valuable details for system configurations as well as potential vulnerabilities. Firewalls and authentication are generally useful in protecting unauthorized access to the systems but fail to monitor the network traffic since most of the attacks arise there. But IDS can block any kind of suspected attacks since it has developed with pre-installed programming regulations.

A wide range of data analysis can be performed by incorporating the log files with the Intrusion Detection System (IDS). The combined system can recognize traffic patterns, anomalies and threats that could not be seen only by the network traffic log file. Intrusion Detection Systems (IDS) can combine security events and details from log files such as IP addresses, timestamps, requests, status codes, URLs, bandwidths and user agents to give a clear view of potentially vulnerable incidents [2]. By this, organizations can easily identify the accurate entries of logs, access the root cause, and protect the suspicious activity.

*Research Question: How are machine learning algorithms used to extract the features and predict the anomalies using the Intrusion Detection System in the log files?*

The main aim of the research is to detect web attacks using Intrusion Detection System(IDS) and classify nine different attacks such as Cross-site Scripting (XSS), SQL Injection, Path Traversal attack, OS command Injection, Carrier Return and Line Feed (CRLF) Injection, Server-Side Includes (SSI) Injection, Lightweight Directory Access Protocol (LDAP) Injection, XPath Injection and Format String attack. Along with this, the paper also examines data preprocessing steps and several features that are necessary for classifying each attack. The implementation is carried out by supervised machine learning algorithms such as Support Vector Machine, Logistic Regression, Random Forest, Gradient Boosting, Decision Tree, and K-nearest Neighbour. The goal is to differentiate the log entries separately using multi-class classification and label them as 'Normal' if no

[2] https://websitesecuritystore.com/blog/what-is-web-application-firewall-in-cybersecurity/

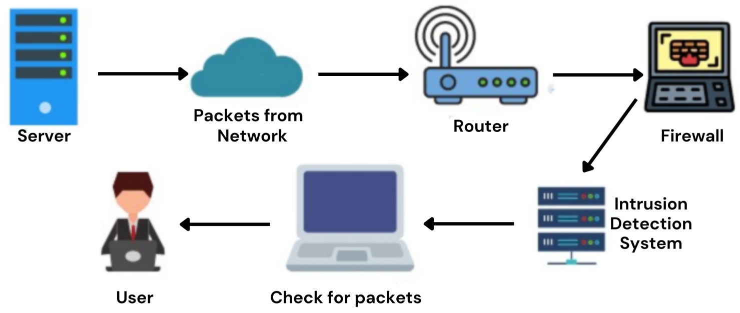

Figure 1: Intrusion Detection System

attacking patterns were matched.

This research paper is segmented into different sections. Section 2 works with the connected part with research done on the same or relevant topic by the researchers. Section 3 is experimented with Research Methodologies and implementation, and these will result in all the technical details of the experimentation. Section 4 discusses the evaluation of Results from the research part resulting in predicting emotions and finding the accuracy of the model. Section 5 will demonstrate conclusions and future work that needs to be done on the topic. The Acknowledgements are written in Section 6 and all the references used in this analysis are given in Section 7.

# 2 Related Work

## 2.1 Intrusion Detection System

Sharma et al. (2020) proposed a system for identifying web attacks based on Intrusion Detection System. This paper mainly focuses on the false positive and false negative prediction scores of the intrusion system. Using the CSIC 2010 HTTP dataset, the main cause of the problem statement is identified. The dataset is produced from the traffic data from the e-commerce web application. The implementation of the system is performed by extracting the necessary 20 features and other pre-processing techniques. The system works on multi-class classification labels such as Cross-Site Scripting (XSS), SQL Injection (SQLi) and Buffer Overflow using machine learning methods such as J48, Naïve Bayes and OneR. The proposed method is evaluated by the Precision score, Recall, Accuracy and F1 Score metrics. From the evaluation, we can conclude that the decision tree classifier has a better accuracy of 94.5%.

Alqahtani et al. (2020) examined the importance of log files by implementing a host-based intrusion detection system. In that system, an agent has been installed in the host, that acts as a transmitter to send and receive data from the server. All of these activities are monitored and managed by the server in the host system which are log files. These data are then checked using rule-set principles that will predict suspicious activities. This paper also concludes that log file entries plays a crucial role in detecting anomalies using HIDS.

Jose et al. (2018) developed a system for finding the deviation and malware attacks in the network. The paper uses KDD Cup 1999 Dataset to compare the performance of six machine learning algorithms such as Bayesian Network, Naive Bayes, Random Forest, Decision Tree, Random Tree, and Decision Table and Artificial Neural Network. The suspicious attacks such as DoS, R2L, Probe and U2R are classified and explained in this paper. Certain drawbacks to be addressed and modified in this paper can be insufficient data and how the model performs and is implemented. Future research is being done on building the data-based intrusion detection system by collecting more appropriate datasets required for the specification

Nasir et al. (2022)combines the several feature selection methods in the NSL-KDD Dataset for intrusion detection. The authors focus on wrapper variants of feature selection algorithms (FSM) which work on swarm intelligence algorithms. The paper proposes a set of feature selection methods which have been four swarm intelligence algorithms (PSO, BA, BAL and BAE) into traditional classifiers (SVM, C4.5 and Naive Bayes). The methods are compared on the NSL-KDD dataset, a large-scale dataset of network intrusion detection. First of all, it presents an extensive examination of FSM and swarm intelligence algorithms. Additionally, it suggests a unique approach by introducing swarm intelligence algorithms into traditional classifiers for feature selection. The authors conclude that their proposed methods can be an aspiring approach to improve the performance of intrusion detection systems. Enough research is not done on different datasets and it has to be tested in real-time applications.

Mazini et al. (2019) presents a different approach of combining swarm intelligence algorithm and meta-learning algorithms such as hybrid artificial bee colony (ABC) and AdaBoost algorithm respectively for anomaly detection. Since ABC algorithm is used for searching the optimistic feature it is combined with the Adaboost algorithm to predict the high accuracy. The combination was compared and executed with other A-NIDSs on the NSL-KDD Dataset. The AdaBoost algorithm is used to classify the network traffic data into normal or anomalous based on the selected features. With feature selections with the ABC algorithm, it was able to handle new features and when required it helped to categorize what is needed for anomaly detection. As expected, the results came out with high precision and accuracy. The paper concluded that it can handle high dimensional data due to the combination of two algorithms. The paper has limitations since it does not compare the algorithm with other parts of A-NIDS. Moreover, it does not discuss the parameters of both algorithms.

Landauer et al. (2023) proposed a novel approach for intrusion detection systems using model-driven engineering principles. It uses the DSML generator tool to automatically generate log data sets. The authors introduced a hierarchical modeling approach to show the topology and types of logs which include system logs, network traffic logs, and application logs. The attacks that are represented are classified as buffer overflows, SQL injection attacks, and denial-of-service attacks. The generated logs are then tested using different intrusion systems which include anomaly-based and hybrid systems. The paper was able to provide real-world scenarios with their prediction. The hybrid approach has given a vast change in evaluating the attacks. The paper limits accuracy as the use of complex tools such as MDE and DSMl can be expensive and not readily available.

## 2.2 Machine Learning Algorithms

Dhanabal and Shantharajah (2015) analyzed the dataset to evaluate the effectiveness of different classification algorithms in network traffic data and the relationship between commonly used network protocols and intrusion attempts. The research was implemented on the effectiveness of decision trees, K-nearest neighbors, support vector machines and K-means. It has been found that decision tree and SVM algorithms were the most effective in detecting network attacks, with both achieving an overall accuracy of 90% from the dataset. It has been proven from this research that TCP has been associated with denial-of-service attacks. This study also concludes that the NSL-KDD dataset is a valuable resource for IDS research, and classification algorithms can be effectively used to detect network attacks based on network traffic patterns. The dataset contained all types of attacks, but it was not able to capture the complex scenarios. Also, there were not enough features to classify the network traffic.

Choudhury and Bhowal (2015) discussed the features and performance of different algorithms in the NSL-KDD dataset and concluded with four categories of attacks. The authors classified the datasets with training and testing sets with various performance metrics which include F-1 Score, precision and recall. From the results, it was found that decision trees and support vectors have better accuracy in comparison with other major algorithms. The dataset was filled with huge records with over 4 million data and was categorized into different attacks which were divided into root and local folders. This paper's findings were useful in developing a more effective IDS system. That study was limited to real-world network traffic data and failed to validate the performance using the cross-validation technique which needs to be performed on the dataset.

Gu et al. (2019)proposed a distinguished approach for intrusion detection using support vectors with feature augmentation. The system applied a logarithm marginal density ratios transformation (LMDRT) approach to all the features in the dataset. The resulting features were trained to detect attacks separating normal and anomalous traffic patterns. The research was conducted on the NSL-KDD Dataset and demonstrated in different categories including denial-of-service (DoS), probing, unauthorized root access (U2R), and unauthorized access to local accounts (R2L). The model was able to predict high accuracy which surpassed traditional SVM methods. The paper also evaluates the algorithm with structured log files and effectively finds the pattern matching and interpretation. This approach was able to identify anomalous patterns effectively. One potential limitation of this theory is the computational complexity of the LMDRT transformation, which can be used for real-time implementation.

Peng et al. (2002)merged classification techniques with feature selections to classify the cyber-attacks. They used a method of vector quantization which was the combination of k-means clustering and correlation feature selection techniques to identify the necessary features. For the classification part, Naive Bayes and decision trees have been used and implemented. It was found that since the hybrid feature selection method was combined with Naive Bayes the accuracy was improved since the focus was on rare cyber threat areas. However only the classification for both rare and common attacks was improved, and the decision method did not work well with feature selection areas. The study was taken in the UNSW-NB15 dataset which was not representative of real-world networking traffic. Also, the paper limits evaluating the performance of algorithms on cross-validation datasets.

Mishra et al. (2019) aims to provide a detailed analysis of intrusion detection using machine learning techniques. The authors examined different machine learning algorithms and their application in intrusion detection. The NSL-KDD dataset was considered and different algorithms like Decision trees, KNN, Support Vectors and Artificial Neural Networks. The research was done and compared with numerous factors which include precision, recall and F1-Score. The authors concluded that through optimizing techniques performance was improved. Decision Trees, SVMs and KNN were algorithms with better accuracy compared to others. This paper focuses on the importance of feature selection techniques which could be better for predicting intrusion detection anomalies. The study was conducted on the NSL-KDD dataset, and it is limited to real-world network representation.

Wenhui and Tan (2001) developed Intrusion Detection System for evaluating the performance of IDSs. The authors proposed the datasets are designed to be reusable and adaptable to different IDS systems. They used the NSL-KDD dataset which contained 4 million records of data that included both normal and attack patterns. For web-based database systems (WBDS), IDSs are particularly important, as they can protect against attacks that target the database itself, such as SQL injection attacks. The Paper used a holistic approach of giving a stage approach which involves an Anomaly Detection Stage and a Machine Learning Stage. This method was able to predict new attacks apart from existing ones from real-world networks. The paper limits conducting research only using a single IDS so the performance lags in real-world performance.

# 3 Methodology

The research methodology will be covered in this section. This section mainly discusses the steps needed to implement the research question technically. Knowledge Driven in the Database(KDD) is used to detect anomalies in the traffic data collected from web applications. KDD is a mandatory step method to apply since it allows for the routine finding of valid, meaningful, and intelligible patterns in large and complicated data sets Corona and Giacinto (2010). The data exploration, data pre-processing, model implementation and evaluation are shown in Figure 2.

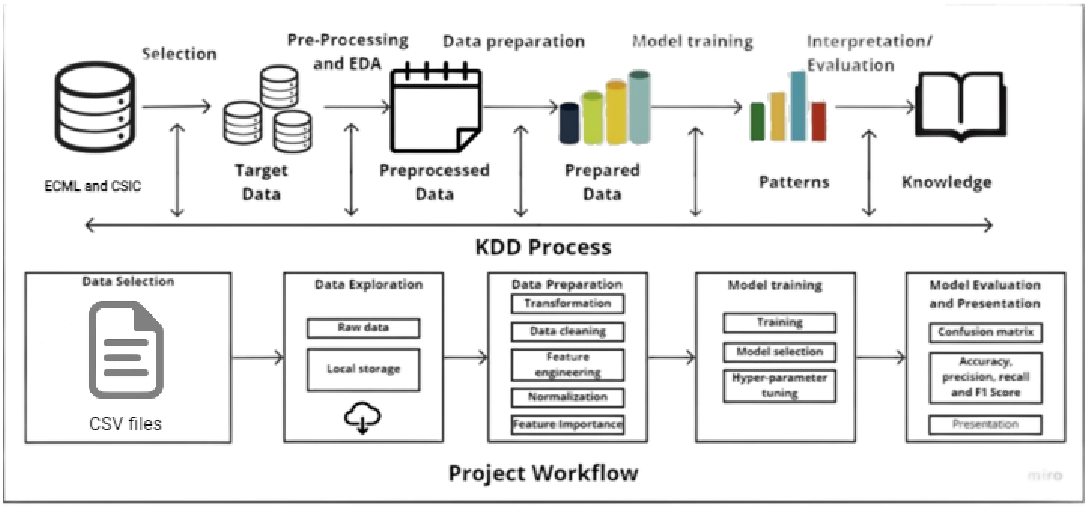

Figure 2: Methodology

## 3.1 Data Selection

Before providing any solution to real-world challenges, it is essential to explore the data and understand the subject of research. Since every subject is peculiar, understanding the intricacies of the domain will aid in addressing the issue and developing a solution. The key issue confronting technology is cyber web-attacks which is considered as the research topic. The increasing number of different types of web attacks was found to be a major concern. This analysis was the initial point of research, after which the exploration of the dataset was performed. This helped the study understand the needs and frame the research question or problem statement based on the requirements.

## 3.2 Exploration of the Data

Finding appropriate data following the project's methodology is the priority at this stage. A file containing .csv files is downloaded from the CSIC dataset from Kaggle and the European Commission of Machine Learning (ECML) open-source website are two files that are useful for the project.

- Log Data (ECML and CSIC): This dataset contains entries of the log files that are prone to web attacks such as IP address, Timestamp, URL, Web attacks, Protocol and Query that have been reported. A CSIC 2010 HTTP dataset is taken which contains the generated traffic containing 36,000 normal and more than 25,000 anomalous requests that include web-based attacks like SQL injection, and cross-site scripting.

## 3.3 Data Preparation

The ECML data contains the details of the log file entries with more than 60000 row items and then the CSIC data is a zip file containing the datasets for various attacks. Initially, all the data exploration was performed in Excel. This was executed mainly because of the merging of required columns and columns related to the problem statement with all the observations before inserting the data into the main data frame. Since the two datasets were collected from different organizations, there are numerous missing values when they are combined and pooled together. The data needs to be processed in a Jupyter Notebook. It is an open-source platform that acts as an interface for complex computing which can be used to execute vast data on single-node machines. As the data deals with network security, it is necessary to have a careful understanding of data pre-processing techniques. The necessary columns that have high priority in detecting web attacks are taken into consideration. Certain columns with some missing values are neglected since it can lead to inaccurate results. By cleaning the dataset and handling certain missing values, new distinct datasets have been created based on the type of attacks. Also, by defining new features, nine different types of datasets were created for the implementation of the research question.

## 3.4 Data Cleaning

For the context of data analysis, once the data is loaded in the data frame all the continuous column values, required categorical columns, and statistical values need to be observed through which data summary can be continued. In data cleaning, handling the null values is the most important procedure. To avoid losing the originality of the data filling null values in all rows and columns wherever needed can preserve the data. The null values in specific columns are identified and then replaced with empty strings. In case null values are affecting the keenness of your analysis dropping the values might be accurate, so certain null values are removed from the subset of rows. On further analysis of data, the fundamental technique of grouping data is carried out by spitting the sets of rows or certain columns which can be valuable while distributing the different categories within the dataset. It allows us to segment the data based on specific criteria, facilitating a more focused analysis of subsets by grouping and counting providing a concise summary of the dataset by removing unnecessary data, making it easier to interpret and communicate findings.

## 3.5 Data Labeling

Data labeling is a crucial step when using supervised machine learning algorithms because these algorithms are trained on labeled datasets. Generally Supervised algorithms learn the input pattern and structure associated with the necessary labels. In this research project, the different types of attacks are considered under multi-class classification labels. The classified labels are important in comparing the truth labels with the predictions.

In this research, we classify nine different web attacks such as Cross-site Scripting (XSS), SQL Injection, Path Traversal attack, OS command Injection, Carrier Return and Line Feed (CRLF) Injection, Server-Side Includes (SSI) Injection, Lightweight Directory Access Protocol (LDAP) Injection, XPath Injection and Format String attack.

- Cross-site Scripting (XSS): Cross-site Scripting is a common type of attack when an attacker feeds vulnerable scripts into the web pages that have been viewed by application users. When these scripts are inserted this type of attack can steal personal data such as login passwords and sensitive data and even cause insecurity to the users.

- Path Traversal attack: Path Traversal attack is a process of accessing the file references and working directories that are outside the root folder and can be manipulated by sequences such as dot dot-dash(../) reference.

- OS command Injection: Also known as Shell Injection, In this attack, the attacker processes the OS commands on the server on the platform in which another application runs. It takes risks application thus progressively affecting other parts of the hardware to other systems within the organization. It leaks the performance of the application and affects the parts of the hardware and this attack spreads to other systems within the organization

- CRLF Injection: CRLF Injection, also referred to as HTTP Response Splitting means “Carrier Return” and “Line Feed”. These characters are present in HTTP response headers which are usually used to represent the “End of Line”. The attacker injects these character sequences in between where they are not expected, which leads to controlling the functionality of the web applications such as session hijacking.

- Server-Side Includes (SSI) Injection: This type of attack is similar to an XSS attack, but the attacker injects the malicious code inside server-side directory files. SSIs are the directives that are present on the web application used to feed the HTML page. The attack comes when user inputs are not validated properly before being used in server files.

- LDAP Injection: LDAP is a Light Weight Access Protocol that is used to give access anyone on a network to access files, individuals and devices. It also happens when user inputs are not tested before being used in queries.
- XPath Injection: XPath Injection is like SQL injection, in which the attacker uses database queries to gain unauthorized access to view the XML documents. It mostly occurs when the user's input is not properly checked in the construction of the query string.
- Format String attack: This type of attack occurs when the submitted data of an input string is evaluated as a command by the application or misuse of format specifiers in programming languages like C and C++.

## 3.6 Feature Selection

Feature selection in predictive analysis is a vital part of machine learning behavior. It selects the relevant features or variables from the subsets and removes the unnecessary input variables. The main advantages of feature selection are its increased accuracy, a reduction in over fitting process and reduced training time. The features in each log vary from system depending upon the application and device. Some of the features selected from the log files are mentioned below: geometry enumitem

margin=1in

| **Field** | **Description** |
|---|---|
| IP Address | 180.163.220.61 |
| Remote Log Name | - (Indicates no specified remote log name) |
| Timestamp | [12/Nov/2019:02:01:00 -0800] |
| Access Request | "GET / HTTP/1.1" |
| Status Code | 200 (Request successful) |
| Bytes Transferred | 12621 (Bandwidth of the page) |
| Referrer URL | "http://www.secrepo.com/" |
| User Agent | "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/50.2.2661.102 Safari/537.36; 360Spider" |

Table 1: Log Information

After doing certain data handling and pre-processing procedures on the basic two datasets ECML and CSIC, distinct datasets were created based on the type of attacks. These distinct datasets have unique features depending on the type of attack. While creating a new dataset, the important step is to label data. The unique keywords that classify each type of attack are defined in lists. These lists are the features of the new distinct dataset created.

## 3.7 Model Training

In the application phase, the selection of the best algorithm or model is known generally as the model selection phase. Many models and different algorithms should be verified against the dataset to predict the best score in terms of metrics. From The selection, the model can be trained and adjusted on the dataset to improve its efficiency. At the initial steps of data analysis, the goal is to predict the degree of injury, which has multiple categories hence, it is classified as multi multi-classification dataset. The first step typically includes such as examining the dataset such as identifying the features present and looking for missing values. The next step is to identify the algorithm for the analysis and choose the best one for the research which could be Random Forest, Boost, Decision trees, K-nearest neighbour, SVM and Logistic Regression.

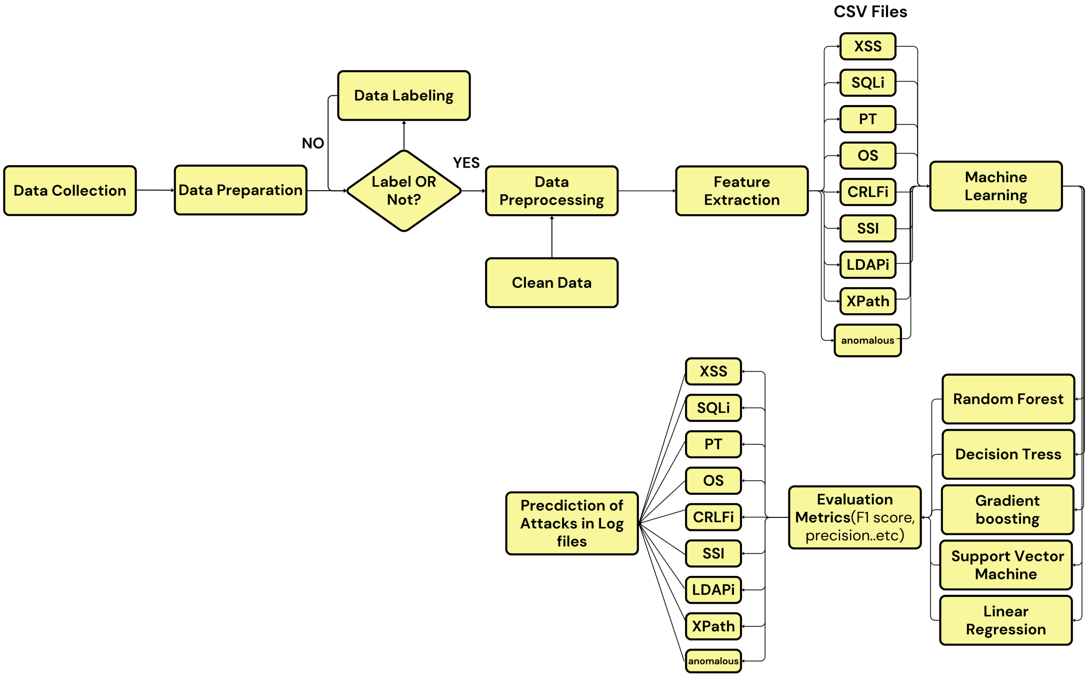

Figure 3: Methodology

# 4 Design Specification

The design specification is an important first step in product management, as it defines the requirements, constraints, and objectives of the machine learning system. It provides detailed information on the methods, algorithms, and expected performance metrics to be used. In this phase, the implementation phase of the process is well handled. The modeling analysis consists of two main steps: select the most appropriate model and apply it to real-world data. The Assessment criteria for each sample are selected based on the research question and preferences.

## 4.1 Modeling Technique

- Decision Tree: The decision tree algorithm is a supervised classification method that utilizes a graphical representation to classify data based on the outcomes obtained when decisions are made using certain conditions. It employs a tree-like structure consisting of root nodes, internal or decision nodes, and leaf nodes. The root node is considered the parent node, while the leaf node is called as child node and serves as an endpoint. The decision tree process begins with selecting a root node and splitting it based on a specific condition, which ends in the creation of a sub-tree. This process is repeated several times until the data is appropriately classified. In Addition to this, pruning can be done to remove unnecessary branches from the tree. To determine the root node, the attribute that best classifies the decision tree, the algorithm calculates the Information Gain (IG). The attribute with the highest IG is chosen as the root nodeRokach and Maimon (2005).

- Logistic Regression: Logistic regression is a supervised regression algorithm used for both classification and regression where the dependent variable (output) can be categorical or discrete. To be more precise the variables are 0/1, True/ False, and Yes/No etc., Logistic regression is a modified version or derived from a linear regression algorithm where a straight line in linear regression is converted into a sigmoid curve also known as S-Curve. The predicted output of logistic regression is either 1 or 0. Here, in the algorithm, a threshold value is being set and if the test value is greater than the established threshold value it is classified as 1, and if the value is less than the threshold value then the resultant is set to 0Peng et al. (2002).

- K- Nearest Neighbours (KNN): K -Nearest Neighbhor is a supervised machine learning algorithm primarily used for classification types. The algorithm is suitable for the datasets where the dependent variable is discrete or categorical such as true/ false, 0/1, yes/no, etc., KNN classifies the data based on the Euclidean distance. KNN classifies new data points by calculating the Euclidean distance between each of them and also to the existing classified data points. By comparing the distances KNN assigns new data points to the classified group among with nearest neighbouring points

- Random Forest: Random forest is a supervised machine learning algorithm that is used for both classification and regression. Here in this Research, random forest is used for classification purposes. The algorithm involves training the model with labeled data and the new data is given for classification. Random forest, as the name suggests it is a forest with a collection of trees. So, a random forest is a combination of decision trees. It is an ensemble model which uses a combination of different supervised machine-learning algorithms. Random Forest breaks a complex problem into smaller ones and solves them resulting in accurate classification results.

- Gradient Boosting: A gradient boosting algorithm is a supervised machine learning algorithm that is used for both classification and regression, which involves the concepts of ensembling. It is a combination of decision trees like that of a random forest, but it builds the decision trees one after the other and computes the results simultaneously. Boosting is usually used to solve complex problems which generally consist of large amounts of data. In many cases, real-world data-driven problems are best classified using boosting algorithms. Boosting is a powerful technique that sequentially builds decision trees to create a strong model.

- Support Vector Machine (SVM):SVM is a supervised machine learning algorithm that is used for both classification and regression. In this project, SVM is used to classify the data using a hyperplane. Initially, we trained the model using label data and later evaluated it on test data. Here, the classification is determined using the hyperplane. A hyperplane is a line that classifies the data, and its dimension is less than its ambient space dimension i.e., if the data is 3-D then the hyperplane is 2-D. To know which is the best hyperplane, we can calculate the distance margin i.e., the distance between the plane and the support vectors where support vectors are the extreme points of two different groups. The plane which has the highest distance margin is considered the best Hyperplane. Now, the new data needs to be classified, taken, and placed so that it belongs to the group on which side of the hyperplane it falls. The hyperplane with the highest distance margin is considered the most favourable. Thus, in this way, SVM classifies the data.

# 5 Implementation

## 5.1 Tools Used

Excel has been used with a combination of machine learning libraries such as Matplotlib and Seaborn to visualize a preliminary understanding of data. Python programming has been used to implement the research question which provides a wide range of data processing, statistical analysis, visualization and algorithms.

## 5.2 Data Selection

The datasets are obtained from open-source platforms such as the European Commission of Machine Learning and CSIC dataset from Kaggle. The dataset contains the details of log entries. By analyzing both datasets, it is determined that these datasets perform better in answering the research question. The URL is the main component where the web attacks are captured in a log file. The URL has two sections namely, the query and the path. Query Analysis helps in understanding the search patterns and browsing functions of the user whereas path analysis provides details of the security and behavior of the user.

## 5.3 Exploratory Data Analysis

To find the pattern and detect the type of web attack, visualization is a mandatory step. Examining the distribution of request types within a dataset can provide valuable insights and serve multiple purposes, depending on the data's context and analysis objectives. The distribution analysis of the type of attacks helps in providing details of the most occurring attacks.

## 5.4 Data Cleaning

From the raw dataset, data cleaning procedures are performed to make the dataset well-suited for implementing the machine learning algorithm. Handling the null values are crucial procedure in pre-processing. From column name "ns1: type" the rows with null values are removed because they affect the type of attack which is the most important

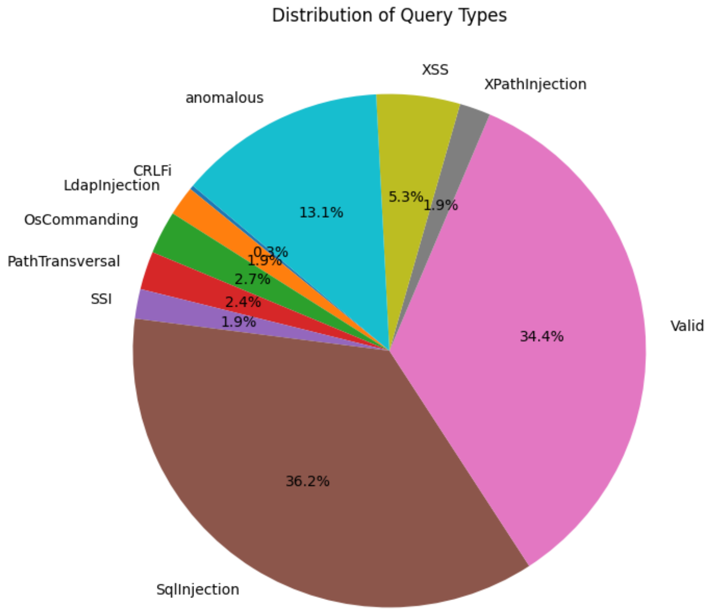

Figure 4: Distribution Analysis of Type of Attacks

column in answering the research statement. In Certain columns such as “path” and “query” the null values are replaced by the empty string. Several columns are renamed for better understanding and analysis of the features to detect the attacks. The CSIC dataset is a zip file that consists of the CSV file several of attacks. So, the main dataset, ECML is changed and converted based on the analysis of the CSIC dataset. The final dataset is formed by combining all the sub-parameters and the ECML dataset. For the final implementation of attacks, new distinct datasets are created.

## 5.5 Feature Selection

The main feature when classifying the type of attack is the URL data from the log file. The log file URL consists of the path and query part of the website. The path and query provide the essential sequence required in classifying the type of attack. The essential features are extracted from the main dataset to form the other new dataset. The new “URL” column is created by concatenating the “query” and “path” columns. The “finaldataset.csv” contains the columns “type ”, and “URL”. The new feature “label” is generated. The next most important analysis is identifying the keywords for each type of attack. There are nine different types of attacks as Cross-site Scripting (XSS), SQL Injection, Path Traversal attack, OS command Injection, Carrier Return and Line Feed (CRLF) Injection, Server-Side Includes (SSI) Injection, Lightweight Directory Access Protocol (LDAP) Injection, XPath Injection and Format String attack. Each attack has unique keywords and patterns to recognize them. Features are extracted from the URL based on the specification of the attacks. For extracting the features for different attacks, the specifications are given in the below code:

- XSS keywords: document ; + ) div var - <script ] ˆ # $ window location search <? — & [ . src cookie iframe createelement string.fromcharcode img / this <>\\, [] ( } onload &# % >: { == eval() onerror ! _ @ = ” ␣* href http .js ’

- SQLi keywords: admin , —— where and between ; ] ˆ <union any exec — into drop ( = like * % not from count () select ␣) commit >= update : replace sleep null - ¿ all + ¡= ¡¿ xp sp delete ’ or insert table /***/ ” . user != char [

- PT keywords: : .bat .conf // system exec winnt \\. ../ ini \\/ :/ windows %00 / ..\\passwd log :\\./ boot etc .. file access ,,

- LDAPi keywords: ; + ) name cn= objectclass =) <— & ) ) ( / \\)& , &( ( = ¿ (— +) ( sn= ! mail = ␣* )) ’

- XPath keywords: ‘ path/ , —— and ; ] ˆ <— ( = comment * % not child () count ␣) position() >= && /* <– - node() >+ <= name <¿ text() // (( or :: user ” . # [

- SSI keywords: dir fromhost httpd winnt\\¡!- replyto .bat etc/ .conf ” date_gmt ”id +connect ”mail +-l –¿ email windows /passwd toaddress access.log var .com +id + virtual #echo bin/ sender # cmd /mail ls+ :\\message +statement #exec home/ odbc log/ system.ini , #include

- OS keywords: ; :\\c: bash .bat shell cmd IP passwd \\/ winnt exec script www. rm ftp — access etc . file -aux :/ .. bin/ ’ wget ..\\\\. ping echo system32 .exe etc/passwd ../ dir log ./ tmp/ display cat root ␣telnet http uname :

- CRLFi keywords: SET %0A %0D + TAMPER : %0D%0A COOKIE

- Anomaly keywords: FU NA FS PL RL FD NK FL AL

- Net Features: dir cmd #exec +id string.fromcharcode sn= winnt access.log %0D%0A + / table onerror mail ¡!- etc/passwd <- .bat union ] // path/ sender :\\node() +connect % href delete ..\\#echo } != =) : & +-l position() admin , where drop insert rm ping cookie ␣comment /mail ” and objectclass location { _ () createelement ../ <>http etc/ virtual log uname <script &( /* .com boot )) %00 fromhost wget access winnt\\any ,, update home/ —— %0A +statement \\¿ ( www. eval() .js :/ )& email — child div [] log/ TAMPER tmp/ ? ’ ; .exe >= like IP name -aux windows ”id bin/ :: date_gmt . \\/ && <– SET &# /***/ # search onload into * <= odbc xp ls+ var all img ˆ file –>= etc this replace c: toaddress .conf ) from root src ) ( document select char system (( ) between display or iframe cat cn= replyto system.ini ! message window exec commit echo $ passwd user system32 not httpd ./ COOKIE ”mail ( count /passwd telnet .. ‘ text() ftp null = @ sp +) ini #include shell script bash == sleep \\ . %0D [ (—

From these keywords, the features are created for each datatype for classifying the attacks. The “label” column is initially assigned to 0. The labels only with the type are set to 1. If the required features are present the in URL, the feature with a specific keyword is set to 1. For example: while preparing the dataset for classifying XSS type of attack, the keywords mentioned above will be considered as the feature of this dataset. Similar steps will be done while creating the other unique distinct dataset. In the figure the label column with “type” VALID is to “0”. In the figure, the same dataset but the label with the corresponding type of value XSS is “1”. This is done to make the model have a better understanding and learning of keywords. This is the most important procedure that aids in providing better accuracy and understanding of the dataset and attacks.

### 5.5.1 Model Implementation

The model is designed to train and test datasets in various machine learning algorithms developed in Jupyter Notebook, which is an environment for programming. First, we can start by importing the required python libraries. The important libraries are Pandas, NumPy, Seaborn, Matplotlib, and Sklearn and the algorithms are DecisionTreeClassifier, Logistic Regression, Support Vector Machine Neighbour Classifier, and Random Forest. Once the dataset is imported the first step is to pre-process all the data to exclude the NaN or missing values. When the data is cleaned and finalized a correlation test needs to be proceeded to establish the relationship between the features. These tests can be processed with built-in function segments such as 'matplotlib' and 'seaborn,' so that the points can be plotted. The next step would be separating the dataset as X and Y with X denoting the independent variables and Y denoting the dependent variables. The model is split into training and testing sets in an 80:20 ratio. The algorithms have been trained across all three algorithms. To calculate the time for models to get trained we can use elapsed time function. The final process will be checking the performance of these classification algorithms using a confusion matrix that represents the values as the true positives, true negatives, false positives, and false negatives. Finally, the precision, recall score f1 score and accuracy of other models are calculated. The above procedures are implemented in the nine different types of datasets generated for each type of attack. Finally, all the models are saved using the ".sav" extension. These files are defined inside a pre-defined function and are used to predict the anomalies in real time. Also, this can categorize the attacks based on the generated log data.

# 6 Evaluation

In the machine learning execution, the model performance and effectiveness are standardized in the evaluation phase. So once the evaluation is performed correctly the model performance will get improved.

## 6.1 Metrics

Accuracy: Shows how many of all predictions were accurate. The number of correct predictions as a percentage is obtained by dividing the total number of predictions, but model performance cannot be fully relied upon for accuracy.

Confusion matrix: The number of predictions per class is specified in this table.

Precision and recall: While recall is the percentage of best models that were expected to be best, precision is the percentage of predicted best models that are best These are often used use in binary classification but can also be used for a wide range of classifications.

F1score: widely used as a statistical measure of classifier performance, and is a consistency factor that improves accuracy and recall.

## 6.2 Discussion

In this research, six machine-learning algorithms were used to classify the type of attacks occurring in the web traffic signals. In the beginning, the model was trained with the

| Attack Type | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| XSS | 0.99 | 0.99 | 0.99 | 24737 |
| SSI | 0.99 | 1.00 | 1.00 | 24721 |
| SQLi | 0.97 | 0.94 | 0.95 | 25196 |
| XPath | 0.98 | 0.95 | 0.97 | 21856 |
| Anomalous | 1.00 | 1.00 | 1.00 | 24742 |
| LDAPi | 1.00 | 1.00 | 1.00 | 24598 |
| PT | 1.00 | 1.00 | 1.00 | 25137 |
| CRLFi | 0.98 | 1.00 | 0.99 | 24526 |
| OS | 0.98 | 1.00 | 0.99 | 23866 |

Table 2: Classification Matrix for Different Types of Attacks using Logistic Regression

ECML dataset, but the model did not yield satisfactory results due to imbalances encountered in the dataset. Then the main ECML dataset was combined with the CSIC dataset to create new distinct datasets to train the model. The results are compared between all the algorithms for all the types of attacks and the best algorithm that classifies the log files is specified for all the attacks. It is observed that KNN has done its best for all the attacks mostly. KNN has best classified the attacks such as Cross-site Scripting(XSS), SQL injection, Path Traversal attack, Server-side Includes(SSI), LDAP injection with an accuracy 98.31%, Gradient Boosting algorithm has best classified the attacks such as OS Command injection attack, LDAP injection, XPath attack with an accuracy 98.57%, Logistic Regression has best classified the XPath Injection attack with an accuracy of 99.56%, Random Forest has best classified the Anomalous attacks with an accuracy of 99.72%, and the attacks such as CRLF Injection and Format String were best classified by all the algorithms with an accuracy of 100% and 99.99% respectively. Thus, by this we conclude that KNN has best classified the web attacks using log files. For all nine types of attacks, the model generated better results by using the machine learning algorithms mentioned above. To access the performance of machine learning model calculating the loss function part is essential. To Achieve high accuracy overfitting is the primary check needs to be verified in the model. Generally overfitting occurs when it works fine with training data but fails to work and generalize with new or unlabeled data. By Determining the relationship within training data and validation we can find the signs of overfitting in the model. When the model constantly runs well on known training data but in turn if it has high loss on the validation set we can opt for overfitting. This specify the model is memorizing the training data instead of learning the data patterns which needs to be applied to new data. Monitoring the loss values on both training and validation sets provides valuable insights into the model's ability to generalize. Checking the loss values in these both sets gives the valuable information about the working of the model.

# 7 Conclusion and Future Work

The aim of this research is to use machine learning algorithms to predict the type of web-attacks of the incoming traffic signals from the log files and explore the analysis of log data. The was collected from the European Commision of Machine Learning wensite and the CSIC dataset from Kaggle. For preliminary analysis, excel is used, and only important columns are taken into consideration. Python library file pandas are used for initial data loading, pre-processing, null values, imputing missing values and perform statistical analysis. Feature selection is the crucial procedure performed by analyssi the keywords for each type of attack. Five machine learning algorithms such as Logistic Regression, Support Vector Machine, K Nearest Neighbor, Decision Tree , Random Forest, Gradient Boost were used to detect the web attacks in log files. The proposed system produced maximum accuracy . To check the overfitting of the model. The loss value for each model calculated was negligible which means the model doesn't overfit and performs better over new data or unseen data. In future research, this work can be expanded by exploring deep learning and reinforcement learning algorithms with different datasets. By implementing the proposed system in an real time environment to explore the possibilities and behavioural analysis.

# References

Alqahtani, H., Sarker, I., Kalim, A., Hossain, S., Ikhlaq, S. and Hossain, S. (2020). *Cyber Intrusion Detection Using Machine Learning Classification Techniques*, pp. 121–131.

Bendovschi, A. (2015). Cyber-attacks – trends, patterns and security countermeasures, *Procedia Economics and Finance* **28**: 24–31. 7th INTERNATIONAL CONFERENCE ON FINANCIAL CRIMINOLOGY 2015, 7th ICFC 2015, 13-14 April 2015,Wadham College, Oxford University, United Kingdom.
**URL:** *https://www.sciencedirect.com/science/article/pii/S2212567115010771*

Choudhury, S. and Bhowal, A. (2015). Comparative analysis of machine learning algorithms along with classifiers for network intrusion detection, *2015 International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM)* pp. 89–95.
**URL:** *https://api.semanticscholar.org/CorpusID:40695073*

Corona, I. and Giacinto, G. (2010). Detection of server-side web attacks, *in* T. Diethe, N. Cristianini and J. Shawe-Taylor (eds), *Proceedings of the First Workshop on Applications of Pattern Analysis*, Vol. 11 of *Proceedings of Machine Learning Research*, PMLR, Cumberland Lodge, Windsor, UK, pp. 160–166.
**URL:** *https://proceedings.mlr.press/v11/corona10a.html*

Dhanabal, L. and Shantharajah, D. S. P. (2015). A study on nsl-kdd dataset for intrusion detection system based on classification algorithms.
**URL:** *https://api.semanticscholar.org/CorpusID:16298036*

Gu, J., Wang, L., Wang, H. and Wang, S. (2019). A novel approach to intrusion detection using svm ensemble with feature augmentation, *Comput. Secur.* **86**(C): 53–62.
**URL:** *https://doi.org/10.1016/j.cose.2019.05.022*

Jose, S., Malathi, D., Reddy, B. and Jayaseeli, D. (2018). A survey on anomaly based host intrusion detection system, *Journal of Physics: Conference Series* **1000**(1): 012049.
**URL:** *https://dx.doi.org/10.1088/1742-6596/1000/1/012049*

Landauer, M., Skopik, F. and Wurzenberger, M. (2023). Introducing a new alert data set for multi-step attack analysis.

Mazini, M., Shirazi, B. and Mahdavi, I. (2019). Anomaly network-based intrusion detection system using a reliable hybrid artificial bee colony and adaboost algorithms, *Journal of King Saud University - Computer and Information Sciences* **31**(4): 541–553.
**URL:** *https://www.sciencedirect.com/science/article/pii/S1319157817304287*

Mishra, P., Varadharajan, V., Tupakula, U. K. and Pilli, E. S. (2019). A detailed investigation and analysis of using machine learning techniques for intrusion detection, *IEEE Communications Surveys & Tutorials* **21**: 686–728.
**URL:** *https://api.semanticscholar.org/CorpusID:65340678*

Nasir, M. H., Khan, S. A., Khan, M. M. and Fatima, M. (2022). Swarm intelligence inspired intrusion detection systems — a systematic literature review, *Comput. Netw.* **205**(C).
**URL:** *https://doi.org/10.1016/j.comnet.2021.108708*

Peng, J., Lee, K. and Ingersoll, G. (2002). An introduction to logistic regression analysis and reporting, *Journal of Educational Research - J EDUC RES* **96**: 3–14.

Rokach, L. and Maimon, O. (2005). *Decision Trees*, Vol. 6, pp. 165–192.

Sharma, S., Zavarsky, P. and Butakov, S. (2020). 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.

Wenhui, S. and Tan, D. T. H. (2001). A novel intrusion detection system model for securing web-based database systems, *25th Annual International Computer Software and Applications Conference. COMPSAC 2001* pp. 249–254.
**URL:** *https://api.semanticscholar.org/CorpusID:42414663*

Zegzhda, D., Lavrova, D., Pavlenko, E. and Shtyrkina, A. (2020). Cyber attack prevention based on evolutionary cybernetics approach, *Symmetry* **12**(11).
**URL:** *https://www.mdpi.com/2073-8994/12/11/1931*