INVESTIGATION OF PERFORMANCE OF MACHINE LEARNING ALGORITHMS OVER CYBER SECURITY DATASETS

Sahil Sharma
Department of Computer Engineering
Pillai College of Engineering, New Panvel
Navi Mumbai, India
ssahil23mtcomp@student.mes.ac.in

Prof. Prashant Nitnaware
Department of Computer Engineering
Pillai College of Engineering, New Panvel
Navi Mumbai, India
pnitnaware@mes.ac.in

ABSTRACT

It is an undeniable fact that currently information is a pretty significant presence for all companies or organizations. Therefore protecting its security is crucial and the security models driven by real datasets has become quite important. The operations based on military, government, commercial and civilians are linked to the security and availability of computer systems and network. From this point of security, the network security is a significant issue because the capacity of attacks is unceasingly rising over the years and they turn into be more sophisticated and distributed. The objective of this review is to explain and compare the most commonly used datasets. This paper focuses on the datasets used in artificial intelligent and machine learning techniques, which are the primary tools for analyzing network traffic and detecting abnormalities.

Keywords—ML algorithms over Cyber Security, Random Forest (RF), SVM, Logistic Regression(LR), Decision Tree (DT), Naive **Bayes Algorithm, Boosting.**

I. INTRODUCTION

The development of machine learning-based cybersecurity solutions is crucial for proactively identifying and mitigating emerging threats. By utilizing techniques such as anomaly detection and behaviour analysis, these solutions provide real-time threat intelligence and enhance system resilience against attacks.

As cyber threats like worms, botnets, malware, and denial-of-service (DoS) attacks become increasingly prevalent, effective protection of networks is essential. This project focuses on assessing the effectiveness of different machine learning algorithms applied to DNS query data.

By detecting anomalies indicative of malicious activity, enabling behavioural analysis, and facilitating automated responses, this approach enhances threat detection and optimizes resource allocation. Insights gained will inform stronger security protocols and strategies, ultimately contributing to a more robust cybersecurity posture.

II. LITERATURE REVIEW

This paper includes a comprehensive literature review titled "Investigation of Performance of Machine Learning Algorithms over Cyber Security Data Sets" offers a comprehensive assessment of past studies, methodologies, and Results pertaining to the implementation of machine learning techniques in cybersecurity. It underscores the critical role of threat detection and outlines the obstacles encountered in this domain. The review examines a variety of machine learning methods strategies, including SVM, AdaBoost, Gradient Boosting, and Decision Trees, highlighting their capability in detecting cyber threats. Additionally, it discusses pertinent datasets, standards for measuring the

performance of the model, and emerging trends within the field. The review ultimately seeks to delineate the existing research landscape, identify gaps in knowledge, and point out the significance of the findings.

Evaluating Cyber Security Threats Using Machine Learning Algorithms: This study investigates how machine learning methodologies enhance cyber security strategies by aiding in the recognition of phishing websites and the timely identification of automated attacks. It provides a thorough overview related to machine learning methods relevant to automated cyber security and intelligent data analysis. Furthermore, the article assesses the effectiveness of various machine learning strategies in tackling real-world issues across different cyber application domains.

Detection and Protection against Cyber Security Threats in IoT Using Machine Learning Techniques: This research introduces recent advancements in cyber security frameworks specific to the IoT and proposes a detailed four-tier cyber risk management framework that includes the ecosystem, infrastructure, risk evaluation, and performance layers. The Cyber Risk Assessment Layer is pivotal in identifying and quantifying IoT cyber risks. The primary objective of enhancing IoT cyber security is to minimize risks for both businesses and individual users by safeguarding IoT assets and ensuring data privacy. Machine Learning (ML) techniques are recognized as highly effective tools for combating cyber security threats and bolstering protection efforts. The study additionally examines the literature regarding IoT cyber security threat detection and protection over the last decade, emphasizing the identification of spam, malware, and intrusions through machine learning methods. This a systematic literature review examines the key ML techniques currently used in cyber security threat detection and protection within the IoT sector. Recently, there has been a significant increase in the application of ML techniques aimed at addressing four essential cyber security challenges: intrusion detection, Android malware detection, spam filtration, and malware analysis.

Deviation detection technique based on the algorithms for defending &mitigating IoT cyber risks in a smart city: investigated an assault and abnormality identification technique formulated upon ML techniques (SVM and KNN) for countering as well as reducing IoT cyber security threats in posh cities. However, as the number of intelligent city networks grows, so does the possibility of cyber-attacks and threats. Intelligent city IoT devices are attached with sensors connected to enormous cloud servers, exposing them to harmful attacks and threats. Subsequently, it is desperate to formulate the procedure to stop corresponding attacks and safeguard IoT devices from crash. ML Algorithm for detection of cyber security threats using Logistic Regression This article gave examples of how Machine Learning analytics may be used to improve cyber security monitoring and research the best algorithms for typical cyber threat situations. ML-based analytics serves as an effective means to delivering context derived from learning security occurrences, resulting in a low probability of false-positive security alarms.

III. EXISTING SYSTEM

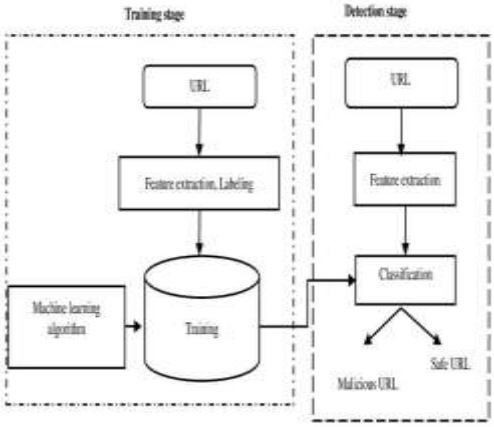

Fig.1. Block Diagram [17]

Malicious URL Detection Model using Machine Learning:

This model operates in two distinct phases: the training phase and the detection phase.

Training Stage:

Initially, to identify malicious URLs, a comprehensive dataset is compiled, consisting of both benign (clean) URLs and those identified as malicious. Each URL within this dataset is meticulously labeled, followed by the identification of crucial features that

differentiate malicious URLs from legitimate ones. In-depth discussion regarding these attributes will be included in this paper. The dataset is then partitioned into two key subsets: one designated for training the ML models and the other reserved for performance evaluation. Should the model showcase satisfactory classification accuracy during the evaluation, it proceeds to the detection phase.

Detection Phase:

During this stage, incoming URLs are subjected to an attribute extraction procedure similar to that in the training phase. The attributes obtained are subsequently analyzed using a classifier, which determines the nature of the URL—whether it is clean or malicious.

IV. PROPOSED SYSTEM

The proposed system utilizes a cybersecurity dataset in CSV format, containing the following columns:

Timestamp: The exact time when the network event occurred.

SourceIP: The IP address associated with the originating source of the network traffic.

DestinationIP: The IP address associated with the destination receiving the traffic.

DnsQuery: Refers to the DNS query that is initiated by the originating source.

DnsAnswer: Represents the response that is returned following a DNS query.

DnsAnswerTTL: Indicates the time-to-live (TTL) value associated with the DNS response, which defines how long the response should be considered valid.

DnsQueryNames: Consists of the domain names associated with the DNS queries made.

DnsQueryClass: Specifies the classification of the DNS query, such as 'IN' for internet queries.

DnsQueryType: Denotes the specific type of the DNS query, examples of which include 'A' for IPv4 addresses, 'AAAA' for IPv6 addresses, and 'MX' for mail exchange records.

NumberOfAnswers: The total number of responses produced in relation to the DNS query.

DnsResponseCode: A code that conveys the status of the response to the DNS query, indicating success or the nature of any error encountered.

DnsOpCode: The operation code that specifies the kind of operation being performed in the DNS message.

SensorId: An identifier for the sensor that captured the network data.

sus: A label indicating whether the observed event is suspicious.

evil: A binary label categorizing events as benign or malicious.

This dataset will be utilized to analyse network traffic and detect potential security threats. The system will implement ML algorithms to classify the data and assess performance using accuracy metrics and confusion matrices. These evaluations will provide insights into the effectiveness of the model in identifying malicious activities and improving cybersecurity defences.

ML models:

A. Logistic Regression:

As a supervised machine learning method, logistic regression is utilized for tasks involving classification, predicting the probability that an instance belongs to a particular class. It analyses the interaction among independent variables and a binary dependent variable, using the sigmoid function to produce probabilities between 0 and 1. If the output exceeds 0.5, the instance is classified as Class 1; otherwise, it is Class 0. The term "regression" reflects its foundation in linear regression, although its primary use is for classification.

B. Naive Bayes Algorithm:

The Naive Bayes a classifier functions as a supervised learning algorithm that predicts class membership based on probability. Named after Bayes' theorem, it presumes that all features are independent from one another. The probability of an event occurring is calculated using the conditional probability of other related events.

C. Support Vector Machine (SVM):

Support Vector Machine (SVM) is a flexible supervised learning algorithm that is utilized for tasks involving both classification and regression. It effectively identifies linear and nonlinear relationships in data, with the goal of identifying the best hyperplane that differentiates the input data into separate classes.

D. Decision Tree (DT):

Decision Trees (DT) are supervised learning models that handle both classification and regression. They construct a tree-like structure by recursively splitting the dataset based on feature values, where each path represents a specific decision.

E. AdaBoost Model:

AdaBoost is an ensemble method that integrates several weak classifiers to form a robust one. Models are built sequentially, with each new model focusing on correcting the

errors of its predecessor, continuing this process until achieving the desired accuracy or reaching a specified limit of models.

V. METHODOLOGY

5.1 Dataset Collection and Preparation:

Gather DNS query logs from tools for network monitoring and systems for intrusion detection, or security appliances. The dataset must contain attributes like a timestamp, query_type, response_code, source_ip, query_name, query_length, and a label indicating if the query is suspicious (1) or non-suspicious (0). Ensure the data is in a structured format (e.g., CSV) and pre-process it by cleaning, addressing missing data, transforming categorical variables, and standardizing numerical features.

5.2 Exploratory Data Analysis (EDA):

Perform EDA to understand the dataset's characteristics. Analyse feature distributions and the target variable, visualize correlations, and identify patterns. This step aids in feature selection and understanding relationships that could improve model performance.

5.3 Model Selection and Training:

Select various machine learning algorithms for classification, such as Logistic Regression, SVM, Decision Tree, Naive Bayes, and Adaboost. Split the dataset into training (80%) and testing (20%) sets, then train each model while tuning hyper parameters for optimal performance.

5.4 Model Evaluation and Comparison:

Evaluate the models using metrics like accuracy, precision, recall, F1-score, and AUC-ROC. Generate confusion matrices to visualize performance and compare measurements to determine the most efficient algorithm for classifying DNS queries.

Deployment and Future Work:

Deploy the best-performing model for real-time DNS traffic monitoring. Continuously update the model with new data to enhance accuracy. Document the methodology and findings, and outline future work, such as exploring advanced techniques for improved detection capabilities.

VI. RESULT AND DISCUSSION

Fig.2. Work Flow

The primary objective of this system is to generate predictions by utilizing a variety of ML algorithms on different datasets. Techniques such as Logistic Regression, Decision Trees, SVM, Random Forests, and Boosting methods will be applied to the DNS query dataset to ascertain which algorithm produces the most favorable outcomes.

Key Components:

Data Collection:
The DNS traffic dataset will be collected in a CSV format, comprising various attributes related to both DNS queries and their responses.

Data Pre-processing:
To tackle class imbalance within the dataset, techniques such as oversampling, undersampling, and synthetic data generation will be utilized, ensuring that each class is sufficiently represented.

Data Analytics:
Feature distributions and interrelationships will be analysed to uncover patterns and correlations, aiding in the selection of appropriate models and enhancing predictive accuracy.

Train-Test Split:
The dataset will be divided into distinct training and testing subsets to facilitate the training of ML models and the evaluation of their performance.

Machine Learning Models:

Various algorithms, including Logistic Regression, will be applied to the training data to classify DNS queries as benign (Class 0) or malicious (Class 1).

Evaluation:

The performance of the models will be evaluated the testing dataset through metrics including accuracy, precision, recall, and F1-score to gauge their effectiveness.

Performance metrics, including the F1-score, accuracy, precision, and recall, are essential for effectively comparing the algorithms. These metrics are particularly significant in fraud detection contexts, where the datasets often demonstrate class imbalance, meaning that instances of fraud are substantially fewer than genuine cases.

1. Accuracy: The percentage of overall predictions that were appropriately categorized as either fraud or non-fraud.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

FP denotes False Positives (Legitimate transactions wrongly detected as fraud)

FN denotes False Negatives (Fraud not detected)

TP denotes True Positives (Fraud detected correctly as fraud)

TN denotes True Negatives (Legitimate transactions detected correctly)

2. Precision: The ratio of properly discovered fraud transactions (also known as positive predictions) to all projected positives. Transactions are expected to be fraudulent.

$$\text{Precision} = \frac{TP}{TP+FP}$$

3. Recall: It measures the percentage of all real positive instances (all actual fraud cases) that are successfully recognized as positive.

$$\text{Recall} = \frac{TP}{TP+FN}$$

4. F1-Score: It is defined as harmonious average of accuracy and recall. Which optimizes stability between accuracy and completeness, particularly when prioritizing one over the other.

$$\text{F1} = 2 \text{ x } \frac{Precison*Recall}{Precisoin+Recall}$$

The dataset produces diverse results when run through several algorithms. Initially, we apply the LR model to the dataset, and results are shown below.

	precision	recall	f1-score	support
0	0.96	1.00	0.98	50
1	1.00	0.50	0.67	4
accuracy			0.96	54
macro avg	0.98	0.75	0.82	54
weighted avg	0.96	0.96	0.96	54

Fig.1. Output for LR

Now the dataset is applied for SVM, and the consequences are shown below:

	precision	recall	f1-score	support
0	0.96	1.00	0.98	50
1	1.00	0.50	0.67	4
accuracy			0.96	54
macro avg	0.98	0.75	0.82	54
weighted avg	0.96	0.96	0.96	54

Fig.2. Output for Support Vector

Furthermore, the dataset was utilized to train a Decision Tree (DT) classifier, and the corresponding results are presented below:

	precision	recall	f1-score	support
0	0.98	1.00	0.99	50
1	1.00	0.75	0.86	4
accuracy			0.98	54
macro avg	0.99	0.88	0.92	54
weighted avg	0.98	0.98	0.98	54

Fig.3. Output for DT model

Furthermore, the dataset was utilized to train a Naïve Bayes Algorithm classifier, and the corresponding results are presented below:

	precision	recall	f1-score	support
0	0.96	1.00	0.98	50
1	1.00	0.50	0.67	4
accuracy			0.96	54
macro avg	0.98	0.75	0.82	54
weighted avg	0.96	0.96	0.96	54

Fig.4. Output for Naïve Bayes

Furthermore, the dataset was utilized to train a Adaboost classifier, and the corresponding results are presented below:

	precision	recall	f1-score	support
0	0.98	1.00	0.99	50
1	1.00	0.75	0.86	4
accuracy			0.98	54
macro avg	0.99	0.88	0.92	54
weighted avg	0.98	0.98	0.98	54

Fig.5 Output for Adaboost

To train and evaluate the ML models, an 80-20 train-test split is employed. Below is a summary of the performance:

Model	Precision	Recall	F1-Score	AUC-ROC
Logistic Regression	0.96	1	0.98	0.99
SVM	0.96	1	0.98	1
Decision Tree	0.98	1	0.99	0.88
Naive Bayes Algorithm	0.96	1	0.98	1
Adaboost	0.98	1	0.99	1

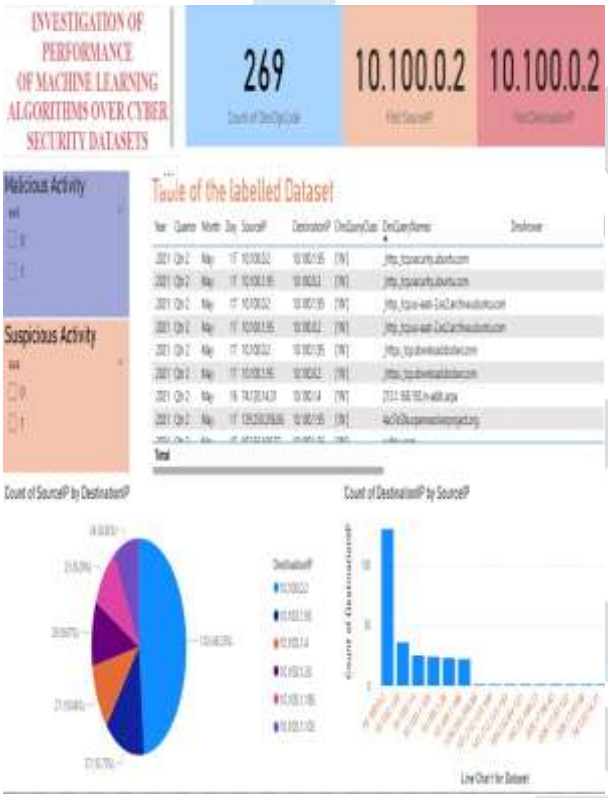

Fig.8. Dashboard for ML-centric CCF Prevention

This Power BI dashboard analyses DNS query data to assess malicious and suspicious activities in cyber security. It features filters for malicious and suspicious activity, with a total of 269 DNS operation codes recorded. The dataset includes a detailed table of queries, highlighting the most active source IPs, notably 10.100.0.2. Visualizations such as pie charts and bar graphs illustrate the distribution of source and destination IPs, while a line chart tracks trends over time. This comprehensive analysis aids in identifying potential threats and informs future machine learning model development for enhanced cybersecurity.

Table 1.1 Comparison chart of different Machine Learning Algorithms

Algorithm	Strength	Weakness	Note
Logistic Regression	1. Easy to implement and interpret. 2. Provides probabilities for class membership. 3. Works well when the relationship between features is linear.	1. Assumes a linear relationship; may underperform with complex patterns. 2. Sensitive to outliers. 3. Limited in handling interactions between variables without feature engineering.	Best suited for datasets where relationships are expected to be linear and where interpretability is crucial.
Naive Bayes	1. Fast to train and predict, especially with large datasets. 2. Performs well with high-dimensional data and categorical features. 3. Simple and effective for text classification tasks.	1. Assumes independence between features, which may not hold true in many cases. 2. Can struggle with rare events if not enough data is available for some classes.	Ideal for applications like spam detection or when dealing with large amounts of categorical data.
SVM	1. Effective in high-dimensional spaces and works well with non-linear boundaries using kernels. 2. Robust against over fitting, especially in high-dimensional data.	1. Computationally intensive, particularly with large datasets. 2. Requires careful tuning of parameters (e.g., kernel type, C parameter). 3. Less interpretable compared to simpler models.	Suitable for complex datasets where patterns are not linearly separable but may require significant computational resources.
Decision Tree	1. Highly interpretable, easy to visualize and understand decision paths. 2. Can capture non-linear relationships and interactions between features. 3. Handles both numerical and categorical data well.	1. Prone to over fitting, especially if not pruned properly. 2. Sensitive to small variations in the data, which can lead to different tree structures.	A good starting point for understanding feature importance and decision-making in the dataset.
Adaboost Model	1. Combines multiple weak learners to create a strong predictive model, improving accuracy. 2. Focuses on misclassified instances, enhancing performance on difficult cases. 3. Generally performs well on imbalanced datasets.	1. More complex and less interpretable than individual models. 2. Can be sensitive to noisy data and outliers. 3. Longer training time due to the sequential nature of boosting.	Excellent choice for achieving high accuracy on complex datasets, but consider using explain ability tools to interpret results.

Deep Learning:

Deep learning can classify DNS queries as suspicious or non-suspicious using neural networks that automatically extract relevant features from query logs. This method achieves high accuracy in detecting malicious activity, though it requires large labelled datasets and substantial computational resources. Overall, it enhances network security by identifying threats in real-time.

VII. CONCLUSION

The proposed system effectively applies multiple machine learning algorithms to analyse DNS query data for threat detection. By systematically evaluating and comparing Logistic Regression, Support Vector Machine, Decision Tree, Naive Bayes, and Adaboost, the system aims to enhance the robustness and accuracy of malicious activity detection in DNS traffic. This multi-algorithm approach can provide valuable insights into the effectiveness of different classification techniques, supporting improved cybersecurity measures. Future work may involve optimizing model parameters and exploring additional algorithms to further enhance detection capabilities.

VIII. FUTURE WORK

Building on the foundations laid by this project, future research will explore the integration of advanced machine learning techniques, including deep learning and reinforcement learning, to enhance threat detection accuracy. Additionally, studies will investigate the application of these models in real-world environments and their adaptability to evolving cyber threats, ensuring continuous improvement in cybersecurity measures.

IX. ACKNOWLEDGEMENT

We want to convey our sincere gratitude to the academic members of Pillai College of Engineering's Computer Engineering Department for giving us the chance and steadfast assistance to prepare this work. Their enthusiastic involvement and knowledgeable direction were essential in enabling this work.

X. REFERENCES

[1] Cho Do Xuan1 , Hoa Dinh Nguyen. Malicious URL Detection based on Machine Learning, 2020.

[2] Dr. Abhilash Maroju, Dr. Srinivas A Vaddadi, Sravanthi Dontu, Analysis on Various Cyber Security Threats on the Basis of Machine Learning Algorithms, 2023.

[3] MS. PRAGATI RANA, DR. B P PATIL, CYBER SECURITY THREATS DETECTION AND PROTECTION USING MACHINE LEARNING TECHNIQUES IN IOT, 2023.

[4] Dr. Rashid khan, Deviation detection technique based on the algorithms for defending &mitigating IoT cyber risks in a smart city, 2021.

[5] Dr. Hari Gonaygunta, MACHINE LEARNING AL CHINE LEARNING ALGORITHMS FOR DE THMS FOR DETECTION OF CYBER TION OF CYBER THREATS USING LOGISTIC REGRESSION, 2023.

[6] Abdel Wedoud Oumar, Peter Augustin D, Credit Card Fraud Detection Using Machine Learning Algorithm, 2019.

[7] DUA, Sumeet; DU, Xian. Data mining and machine learning in cybersecurity. CRC press, 2016.

[8] Canongia, Claudia, and Raphael Mandarino Jr. "Cybersecurity: The New Challenge of the." Handbook of Research on Business Social Networking: Organizational, Managerial, and Technological Dimensions: Organizational, Managerial, and Technological Dimensions (2011): 165.

[9] TWOMEY, Paul. Cyber Security Threats. 2010.

[10] Von Solms, Rossouw, and Johan Van Niekerk. "From information security to cyber security." computers & security 38 (2013): 97-102.

[11] Fraley, James B., and James Cannady. "The promise of machine learning in cybersecurity." SoutheastCon, 2017. IEEE, 2017.

[12] https://www.symantec.com/content/dam/symantec/docs/otherresources/webapplicationfirewall-owasp-top-10-2017-coverage-en.pdf

[13] Buczak, Anna L., and Erhan Guven. "A survey of data mining and machine learning methods for cyber security intrusion detection." IEEE Communications Surveys & Tutorials 18.2 (2016): 1153-1176. 27

[14] Thuraisingham, Bhavani, et al. "Data mining for security applications." Embedded and Ubiquitous Computing, 2008. EUC'08. IEEE/IFIP International Conference on. Vol. 2. IEEE, 2008.

[15] Meshram, Ankush, and Christian Haas. "Anomaly detection in industrial networks using machine learning: a roadmap." Machine Learning for Cyber Physical Systems. Springer Berlin Heidelberg, 2017. 65-72.