Grenze International Journal of Engineering and Technology, June Issue

Efficient Classification of Anomalous Activities in Web-based IoT Systems using Feature Reduction Techniques

Shraddhanjali Sahoo[1], Sucheta Panda[2] and Sasmita Acharya[3]
[1-3]School Of Computer Sciences, VSSUT, Burla, India
Email: shraddhanjaliphdca23@vssut.ac.in, suchetapanda_mca@vssut.ac.in, sacharya_mca@vssut.ac.in

***Abstract*—The adoption of IoT has become important due to the effortless access and remote management of things in the environment. This is made possible through the integration of web-based systems into IoT, allowing accessibility through the internet but due to this, there is always a chance get being targeted by the attacker by finding the security vulnerabilities in the edge or the cloud-based integration. In this work, We present a model designed to classify anomalous activities within a web-based IoT system. Our approach encompasses binary classification, achieved through an efficient reduction of the feature space. through this, our system enhances interpretability and optimizes model efficiency, ensuring precise and effective classification. To obtain a condensed feature subset, our method combines Pearson correlation analysis with permutation-based feature importance. We assessed several machine learning models, including Logistic Regression(LR), K-Nearsest Neighbour(KNN), Random Forest(RF), and Xgboost(XGB), for binary classification tasks using the CICIoT2023 dataset. With just 10 selected features, Xgboost and Random Forest among these models frequently produced outstanding results, claiming 99% accuracy and a 98% F1 score. This demonstrates how successful and efficient the method is at categorizing unusual IoT Web-based activity.**

***Index Terms*— IoT, Web-based attack, machine learning, Feature selection, Classification.**

I. INTRODUCTION

The Internet of Things (IoT) is a pillar of the digital age, revolutionizing daily activities and business processes alike. With seamless connectivity, IoT devices collaborate to create a smarter, more interconnected world. However, this connectivity also introduces vulnerabilities, particularly in web-based interactions [1]. The integration of IoT technology marks a significant milestone in various domains, facilitating enhanced communication and automation. IoT ecosystems leverage sensor data, real-time analytics, and cloud computing from smart homes to autonomous vehicles to streamline operations and improve efficiency. Yet, this proliferation of interconnected devices comes with a stark reality: heightened susceptibility to increased risk of security breaches and data compromises. As devices rely on web-based programs for control and communication, they become prime targets for various forms of web target attacks. The potential assault may aim at various integral components of IoT infrastructure, spanning from the intricate firmware embedded within devices, the user-facing web interfaces, and the underlying communication protocols facilitating data exchange, to the pivotal cloud services and APIs entrusted with data storage and management. These attacks come in

Grenze ID: 01.GIJET.10.2.656

different forms, such as flooding a website with too many requests (HTTP Flood), sneaking into databases to steal data (SQL Injection), secretly injecting malicious code into web pages (Cross-Site Scripting), and taking control of web browsers (Browser Hijacking)[2]. Each attack is dangerous and needs strong defenses to stop them. In the realm of IoT security, traditional methods like Web Application Firewalls (WAFs) and authentication mechanisms fall short in effectively countering sophisticated web-based attacks, The challenge lies in the ability of attackers to mimic benign traffic patterns, rendering conventional detection methods inadequate [3]. To address this, we propose an approach—a real-time dataset-driven methodology aimed at designing a machine learning model tailored for detecting web-based attacks in web applications-based IoT systems, By leveraging diverse machine learning algorithms and implementing efficient feature selection techniques, our objective is to improve detection precision and strengthen cybersecurity protocols within IoT ecosystems.

The further sections of the article are organized as follows: In Section II discusses related work done in the area, Section III demonstrates the proposed model, Section IV shows the Result with the analysis of the Experiment and Section V Concludes the paper with Future Work.

II. Related Work

The integration of IoT systems increases because of their efficiency in controlling the various aspects of modern life which increases the vulnerability to in great extent. while Tarek Gaber et al. [5] introduced a novel intrusion detection system for binary classification that was created explicitly to address SQL injection risks in smart cities. The 154 features are reduced to just 8 features through the use of constant removal and recursive feature elimination feature selection algorithms. Decision trees outperformed other popular machine learning models in tests conducted against them, achieving an astounding 99% accuracy on the AWID dataset. Chaochao Luo et al. [6] propose an ensemble of three deep learning models, namely MRN, LSTM, and CNN, for web-based attack detection. The models were trained on publicly available datasets, such as the HTTP CSIC dataset 2010, and datasets collected in real-time by other researchers using tools like Burpsuit, SQL Map, and others. The models achieved the highest performance, with an accuracy of 99% with a True positive rate of 99%. To overcome the shortcomings of conventional intrusion detection systems and safeguard numerous web applications on the edge, Zhihong Tian et al. [7] introduced distributed deep learning for attack detection. This method is centered on analyzing URLs for deployment at edge devices. that employ numerous concurrent deep learning frameworks using CNN and Resnet. and they were contrasted with several datasets, including CSIC 2010, where M-Resnet with word2vcc achieved 99.41% accuracy and 98.91% True positive rate. Prince Roy et al. [8] use several state-of-the-art methods, such as logistic regression, Naive Bayes, AdaBoost, Random Forest, and XgBoost to find a major server vulnerability. Through the identification of logical errors and the capability to perform a SQL Injection Attack against web-based systems, Naive Bayes achieves 98.33 accuracy, and 100% sensitivity than other ML models. Using deep learning techniques, Tekerek et al. [12] presented an architecture for anomaly-based web attack detection in a web application Using Convolution neural network (CNN) architecture, they considered Numerous web application attacks, including SQL injection, buffer overflow, CRLF injection, XSS, and parameter tampering and this approach tested on the CSIC2010v2 datasets which gives the detection accuracy of 97.07% and the F1 score with 97.51%. Recently S. Jaradat et al. [4] presented a hybrid model in their work that was designed to categorize web-based attacks as either malicious or benign. They optimized the parameters and feature selection by combining the Genetic algorithm with the Tree-based Pipeline Optimization Tool (TPOT) where Gradient boosting was the most successful classifier among the several that were tested, including logistic regression, MLP, and KNN. It achieved an astounding 95% accuracy, 94% recall, and a 95% F1 score on the CICIoT2023 dataset.

III. Methodology

This section outlines our approach to web-based attack classification in the IoT environment. We begin by discussing the dataset chosen for our study, emphasizing its relevance. We then detail our framework in Fig.2 which prioritizes efficient feature selection to get the optimal no of features. Following this, we train multiple machine learning models using the selected features and evaluate their performance in classifying web-based attacks. This systematic approach aims to identify the most effective features for ML algorithms to get accurate attack classification.

A. Dataset

The dataset utilized in this study is CICIoT2023 provided by the Canadian Institute for Cybersecurity. It serves as a benchmark real-time dataset for attack detection in IoT environments. This dataset encompasses 33 different categories of attacks, alongside benign traffic, with 46 features in CSV format. For our study, we extracted both benign traffic and various attack categories, including DDOS- HTTP Flood, DOS, HTTP, Flood Slowloris, Dictionary Attack, Browser Hijacking, SQL Injection, Command Injection, Uploading Attack, and Cross-Site Scripting (XSS). These attacks can target web resources in different forms. The extracted data from this dataset serves as the basis for evaluating our proposed approach with different well-known ML models [9]. Fig.1 displays the distribution of samples for binary classification. It is observed that attack categories have less data than benign attack. In this case, careful consideration of model deployment and feature selection is essential so that the model should not be biased towards the majority class.

B. Preprocessing

To improve the machine learning model's prediction accuracy by using more pristine, error-free data. To do this, we must handle the missing values and eliminate duplicate and null values from the data. Furthermore, the class label is encoded through label encoder which encodes the categorical label to a numeric form Additionally, to enhance model interpretability, prevent feature dominance, and mitigate overfitting during training, we have implemented min-max scaling. This technique transforms data to a range between 0 and 1. The (1) shows the representation:

$$X' = \frac{(X - X_{MIN})}{X_{MAX} - X_{MIN}} \qquad (1)$$

Where X' represents the scaled feature value of X, X_{MIN} minimum feature value of X and X_{MAX} is the maximum feature value in the X in the dataset.

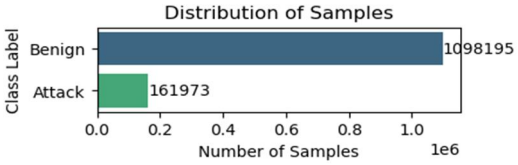

Figure 1. Classwise Data Distribution for the Model

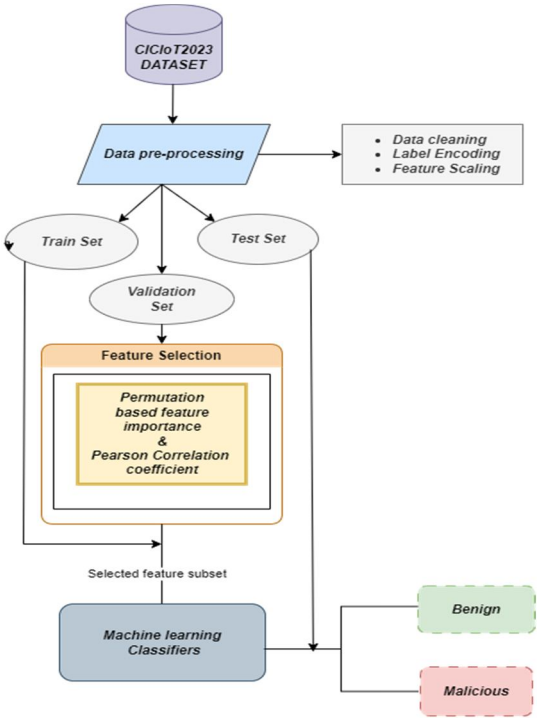

Figure 2. Detailed Approach on Classifying vulnerabilities for Web-Based System

C . Feature Selection

When deploying a machine learning model, it's crucial to train it on meaningful features while filtering out extraneous information that could compromise its performance. Additionally, reducing the model's complexity and minimizing early response to real-world data. To achieve this in the proposed work, the permutation-based feature importance method is used which is model-independent and is robust against the biased nature present in the conventional approaches. This method ranks features according to the drop in the ML model performance. It involves shuffling each feature K time in the feature set $X \in D$, where X contains $i = 1 \dots n$ features. Features that show high drops in performance after being shuffled are thought to be crucial for correctly identifying the target classes[10].

$$f(i) = P\{m(X_i, Y)\} - \frac{1}{K} \sum_{K-1}^{K} P\{m(X_i' , Y)\} \qquad (2)$$

In Equation (2) $f(i)$ represents the importance of feature i, calculated as the difference between the model's performance before and after permuting the feature X_i with respect to the predicted variable Y . In our proposed model implementation, we train classifiers and test it using the validation set to determine feature importance based on the mean decrease in F1-score because of the imbalanced nature of the dataset, this plays a vital role in assessing model performance. Features are ranked based on their importance scores, and only those within a defined N no features are selected. Fig .3 displays the feature importance of each feature in the graph by the best-performing model.

Further, we employ the Pearson correlation coefficient (PCC) that is used to find the linear correlation between the features to eliminate the redundancy. If the correlation coefficient is close to 1 0r -1, it indicates a strong relation, if it's close to 0,

it shows no correlation [13]. To obtain this we employ to find the correlated feature set above a threshold and remove the feature according to the rank achieved by the feature importance selecting the best possible feature to maintain the high model accuracy. Table I shows the selected features used for model training.

Algorithm-1: Proposed Approach for Classification

Input-: feature set $f(X)$, Target $(Y) \in D$ (Dataset)

Initialization: Model $M \in \{M_1 \dots , M_n\}$

1.split D →D_train , D_val, D_test
2. For Each M_i →Train(M_i) , D_train)
3 Predict Y_i=(M_i,D_val)
4. M_{select}=Max P(M_i,D_val)
5. End For
6. For feature $f(X) \in$ D_val
7. Permute $I(X_i)$←(M_{select})
8. End For
9. For feature $f(X) \in D$
10. Compute Pcc= [corr(X_i , X_j)] > $threshold$
11. Remove X_i , if $I(X_i)$< $I(X_j)$ else Remove X_j
12. End for
13. Select top $X_k' = \{X_1 \dots , X_k\}$
14. Evaluate P_i= Test(M_i, X_k ,D_test)

Output:- Important feature(X_k'), Classification performance of Model $M \in \{M_1 \dots , M_n\}$

TABLE I. SELECTED FEATURE FROM CICIOT2023 DATASET

Selected Feature	Description
IAT	How much time has passed since the last packet
flow_duration	Time spent flowing the packet
HTTPS	Shows the presence of application layer protocol is HTTP
HTTP	Shows the presence of application layer protocol is HTTPS
Header_Length	Length of the header
Rate	The speed at which packets are transmitted in a flow
syn_count	Quantity of syn flag-set packets in a single flow
urg_count	Quantity of urg flag-set packets in a single flow
fin_count	Quantity of fin flag-set packets in a single flow
SSH	Shows the presence of application layer protocol is SSH

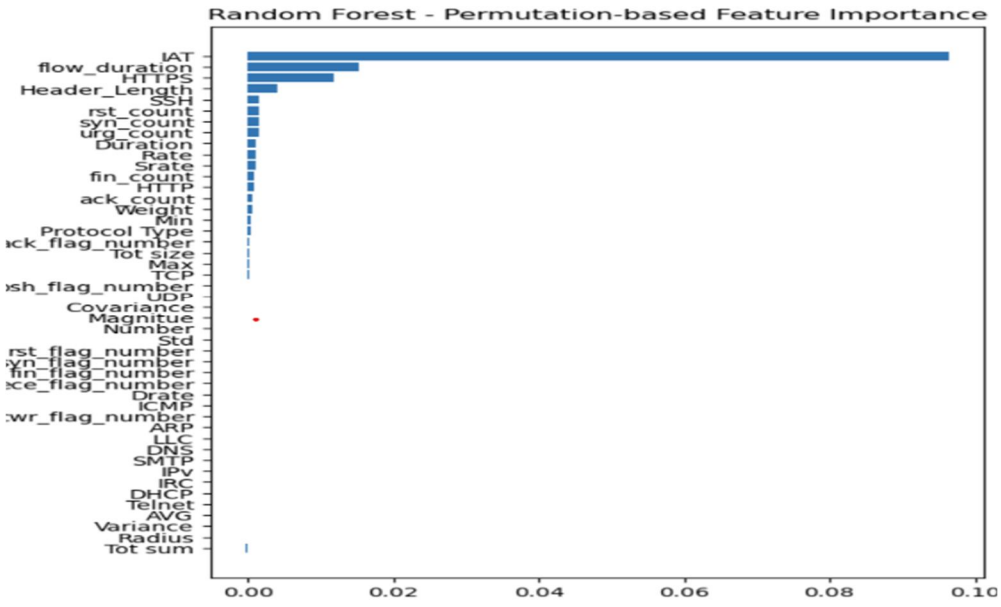

Figure 3. Importance of Each Feature by Random Forest Model

IV. Result And Discussion

This section summarizes the effectiveness of our proposed methodology using several machine learning algorithms in binary classification scenarios. We evaluated the performance of our feature selection method on the CICIoT2023 dataset,
aiming to distinguish web-based attacks and enhance the comprehensibility of dataset complexity. Specifically, we assessed the impact of feature selection on performance in binary classification, where class 0 represents non-attacks and class 1 represents attacks encoded by label encoder. This approach enabled a comprehensive evaluation of the effectiveness of our method across different attack scenarios.

A. Performance Evaluation Metrics

We use a variety of strategic indicators of evaluation that measure the model's performance in appropriately classifying unknown data which include the below formulas utilizing True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN) values [11].
Accuracy: Calculates the percentage of cases that are correctly classified out of all the instances.

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN} \quad (4)$$

Precision: This also called positive predictive value(PPV), highlights the model's ability to avoid false positives by displaying the ratio of accurately predicted positive instances to all predicted positives.

$$\text{Precision} = \frac{TP}{TP+FP} \quad (5)$$

Recall: It indicates how well the model can capture all positive situations by showing the ratio of accurately predicted.

$$\text{Recall} = \frac{TP}{TP+FN} \quad (6)$$

F1-Score: harmonic mean of recall and precision, providing a fair evaluation of the model's effectiveness.

$$\text{F1-Score} = 2 * \frac{Precision*Recall}{Precision+Recall} \quad (7)$$

B. Experimental analysis

The steps utilized in the proposed work are discussed in Section III, which is being implemented with different machine-learning models to classify malicious and legitimate activity within the web-based integration system. In this section, the experiments demonstrate how the minimum number of features selected by the feature selection technique, maintains high performance across different well-known ML algorithms.

In Table II, we present the binary classification report showcasing the evaluation results of various models. Logistic Regression exhibits an accuracy of 91% and an F1 score of 80%. Similarly, KNN achieves 96% accuracy and 93% F1 score. Notably, Random Forest and Xgboost outperform other models, attaining 99% and 98% accuracy respectively, along with 97% F1 score. These results underscore the significance of these models in effectively distinguishing between benign and malicious attacks irrespective of the imbalanced nature of the data.

For a comprehensive understanding of model performance, Fig.4 presents a heatmap of the confusion matrix. This heatmap visualizes the classification accuracy of each model, detailing the performance for each class in terms of correct classifications and misclassifications. Specifically, it shows true positives, false positives, true negatives, and false negatives within our proposed model. The positive class represents benign labels, while the negative class denotes attack labels. Notably, the Random Forest and Xgboost classifier demonstrates high classification accuracy, correctly identifying 99% of the positive class. However, it misclassifies only 4 instances within the negative class. While other models have misclassified the attack class heavily. This detailed breakdown provides a clear insight into the effectiveness of classifier
models.

In Fig. 5 Receiver Operating Characteristic Area Under the Curve or ROC AUC, evaluates a model's capacity to distinguish between classes across all potential thresholds. An improved ability to discriminate is shown by a higher ROC AUC score, which runs from 0 to 1. with 1.0 being the perfect score and 0.5 indicating the random chance [11].

TABLE II. EXPERIMENTAL RESULT OF MODELS ON BINARY CLASSIFICATION

Model	precision	Recall	F1score	Accuracy
LR	0.92	0.71	0.80	0.91
KNN	0.96	0.90	0.93	0.96
RF	0.98	0.96	0.97	0.99
XGB	0.98	0.95	0.97	0.98

It is always preferred over accuracy in imbalanced datasets where the classes are unevenly distributed because it evaluates the model's ability to correctly identify positive instances (true positives) while minimizing false positives, regardless of the class distribution. In our proposed framework, both XgBoost and Random Forest achieved perfect ROC-AUC scores of 1.0. Additionally, Logistic Regression and KNN exhibited strong performance with impressive discriminative capabilities.

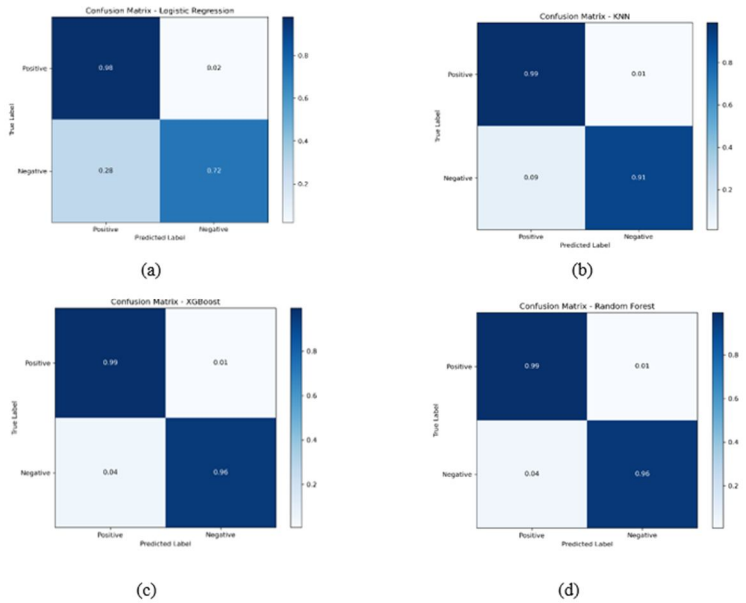

Figure 4. Confusion Matrix of Machine Learning Models

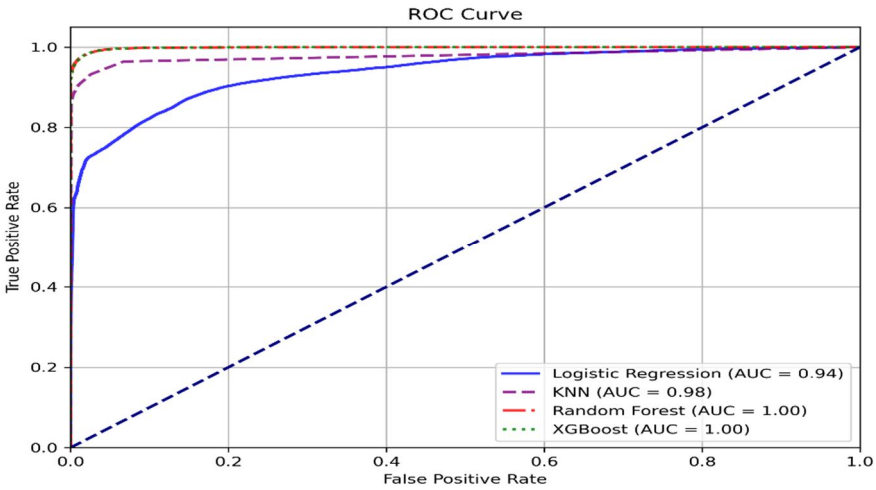

Figure 5. ROC-AUC Result of Different Machine Learning Models

V. Conclusion And Future work

In our study, we aimed to classify vulnerabilities in IoT web integration systems and reduce the dataset's dimensionality for improved model efficiency. By employing a feature selection technique that combines permutation feature importance and correlation analysis, we effectively identified and removed irrelevant features, this dimensionality reduction streamlined the dataset, enabling our models to accurately classify attack data among benign traffic. The integration of feature selection played a pivotal role in enhancing the interpretability and performance of our ML classifiers. It allowed for more focused model training, thereby improving classification accuracy. Moving forward, there is still a need for improvement in cases where imbalanced data is present for training, we need a robust method to handle the imbalanced data and explore different meta-heuristic feature selection techniques to further optimize and implement the approach in different datasets to ensure its efficiency in real-world applications.

References

[1] Y. An, F. R. Yu, Y. He, J. Li, J. Chen and V. C. M. Leung, "A Novel Internet of Things (IoT) Web Attack Detection Architecture Based on the Combination of Symbolism and Connectionism AI," IEEE Internet of Things Journal, 2024.

[2] M. Khari, P. Sangwan and Vaishali, "Web-application attacks: A survey," 2016 3rd International Conference on Computing for Sustainable Global Development, New Delhi, India, pp. 2187-2191, 2016.

[3] R. Lomte, and S. Bhura. "Survey of different Web Application Attacks & Its Preventive Measures," IOSR Journal of Computer Engineering (IOSR-JCE) ,pp. 46-51 , 2013.

[4] A. Jaradat, A. Nasayreh, Q. Al-Na'amneh, H. Gharaibeh and R. Al Mamlook, "Genetic Optimization Techniques for Enhancing Web Attacks Classification in Machine Learning," 2023 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress Abu Dhabi, United Arab Emirates, pp. 0130-0136, 2023.

[5] T. Gaber, A. Ghamry, and A. Hassanien. "Injection attack detection using machine learning for smart IoT applications," Physical Communication, 2022.

[6] C. Luo, Z. Tan, G. Min, J. Gan, W. Shi and Z. Tian, "A Novel Web Attack Detection System for Internet of Things via Ensemble Classification," in IEEE Transactions on Industrial Informatics, vol. 17, no. 8, pp. 5810-5818, Aug 2021.

[7] Z. Tian, C. Luo, J. Qiu, X. Du and M. Guizani, "A Distributed Deep Learning System for Web Attack Detection on Edge Devices," in IEEE Transactions on Industrial Informatics, vol. 16, no. 3, pp. 1963-1971, March 2020.

[8] P. Roy, R. Kumar and P. Rani, "SQL Injection Attack Detection by Machine Learning Classifier," International Conference on Applied Artificial Intelligence and Computing (ICAAIC), Salem, India, pp. 394-400,2022.

[9] E. Neto, S. Dadkhah,R. Ferreira, A. Zohourian, R. Lu, and A. Ghorbani. "A real-time dataset and benchmark for large-scale attacks in IoT environment, " Sensors, (13), pp. 5941, 2023.

[10] A. Fisher, C. Rudin and F. Dominici, "All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously," Journal of Machine Learning Research, pp. 1- 81,2019.

[11] Z. Vujović,."Classification model evaluation metrics," International Journal of Advanced Computer Science and Applications vol 12, Issue.6, pp.599-606,2021.

[12] Tekerek, "A novel architecture for web-based attack detection using convolutional neural network," Computers & Security vol 100, 2021.
[13] M. Hall, "Correlation-based feature selection for machine learning," PhD diss., The University of Waikato, 1999.