Advances in Nonlinear Variational Inequalities
ISSN: 1092-910X
Vol 28 No. 2s (2025)

Optimized Web Server Attack Detection: A Super Learner Ensemble Model Approach

Sainath Patil[1,2*], Rajesh Bansode[3]

[1]Research Scholar, Dept. of IT, Thakur College of Engg and Tech, Mumbai University, Maharashtra, India

[2]Assistant Professor, Dept. of IT, Vidyavardhini's College of Engg and Tech Vasai, Mumbai University, Maharashtra, India

[3]Professor, Dept. of IT, Thakur College of Engg and Tech, Mumbai University, Maharashtra, India

*Corresponding email: psai17@gmail.com

Article History:

Received: *20-09-2024*

Revised: *30-10-2024*

Accepted: *12-11-2024*

Abstract:

Web applications are essential for many organizations, yet they are vulnerable to various security threats, such as injection attacks and inadequate authentication mechanisms. To address these risks, this study proposes a super learner ensemble learning model that combines multiple machine learning (ML) algorithms to improve web server attack detection. Leveraging the unique strengths of each base ML model, the super learner approach enhances predictive accuracy by using a meta-model trained on out-of-fold predictions from base learners, achieving superior performance in identifying attacks. The proposed model was evaluated on the UNSW-NB 15 and KDD CUP 99 datasets, achieving impressive detection accuracies of 99.69% and 99.90%, respectively. This ensemble model effectively addresses challenges in cybersecurity, such as high false-positive rates and imbalanced data, by employing adaptive synthetic sampling and feature selection. Comparative analysis reveals that the super learner model outperforms existing detection methods, improving detection accuracy by up to 9.54%. These findings suggest that the super learner ensemble approach is a promising method for enhancing the security of web applications. Future work could expand on these results by exploring different base models, datasets, and real-time anomaly detection mechanisms to further improve web server protection.

Keywords: Computer Security, Machine Learning, Super Learner Ensemble, Server Security, Web Server Attacks

1. Introduction

For many business organizations, web applications are vital asset as they provide convenient access to a wide range of information and services related to the business. However, these applications are susceptible to exploitation by malicious user due to several common security flaws like, injection vulnerabilities, inadequate authentication mechanisms, and misconfigurations of both the application and web server environments. The malicious user/ attacker often target web servers for a variety of reasons, such as extortion, financial gain, corporate rivalry or sometimes excitement. Implementing

cutting-edge methods for identifying and thwarting attacks on web applications is necessary to counter these risks and improve security [1-3].

The enormous dimensionality of network data offered by the systems that monitor networks nowadays makes it possible to apply machine learning techniques widely to enhance the identification and categorization of unusual events. Though it's widely acknowledged that there isn't a magic solution for solving several issues at once, choosing the optimal machine learning model for a given situation is a challenging undertaking [4, 5].

Finding a model that performs best for various data distributions and statistical combinations may be quite challenging, even while there may be several that are very well suited for a given task. It is possible to combine several models to create a better one using the ensemble learning theory. In order to achieve greater predictive performance than could be attained from any one of the machine learning algorithms alone, ensemble approaches employ numerous learning algorithms. Moreover, an ensemble of models shows improved resilience to training data uncertainty, which is very advantageous [6].

In comparison to single-based learning methods, ensemble learning theoretically has a larger computing cost and complexity. That being said, this limitation is lessened by the fact that modern big data platforms and off-the-shelf data processing technology are developed sufficiently to support the quick and parallel execution of numerous algorithms.

In this paper a super learner ensemble learning model is proposed to detect attacks on web server. The super learner approach uses several ML models referred as base learners applied on k-fold data for predictions and a meta model then fitted to the out-of-fold predictions produced by each base learner.

2. Litarature Survey

Various machine learning (ML) techniques have been explored by researchers in the past to identify attacks on web servers [7, 8]. In this part, several research studies are reviewed, and important issues are noted.

To assess the impact of incoming traffic on server performance D. M. Sharif et al used a particular threshold-based sampling approach to sample and analyze incoming traffic to the webserver. Three different thresholds considered to select traffic samples based on the server's resource utilization, viz one-tenth of the incoming traffic is sampled when resource utilized up to 30% of their capacity, one-third sampled when utilization in bet 30-70% and half of the incoming traffic sampled the utilization reaches above 70%. Gaussian Mixture Models (GMM) and Random Forest (RF) classifier used to check samples are normal or malicious. If samples are not normal, then human expertise with distributed denial of service (DDoS) detection inspect the sample to determine indeed DDoS attack. If experts confirmed the attack samples, then the new attack pattern is added to the database. The classifier then re-trained with new database to improve its detection capabilities for future attacks. A robust defense against DDoS attacks is ensured but humans are involved as DDoS expert component to identify new type of attack [9].

The research study presented by Ismail S. et al [10] used Kendall's correlation coefficient and Mutual Information (MI) feature selection techniques independently to select the optimal feature subset. The authors proposed a light-weight ensemble-based ML approach, Weighted Score Selector (WSS) to

detect cyber-attacks on Wireless Sensor Network (WSN). Naive Bayes (NB), k-Nearest Neighbour (kNN), RF, Support Vector Machine (SVM) and Light Gradient Boost Machine (LightGBM) are some of the heterogeneous supervised ML algorithms that were used to construct the proposed light-weight ensemble based WSS model. Proposed approach achieved promising results with respect to detection accuracies considering all_attacks, Grayhole, and Flooding datasets, but the other performance parameters are not considered.

In order to address the high false-positive rates, imbalanced data with poor training performance, low detection accuracy, and complexity in feature selection seen in the majority of ML-based models, G. Mohiuddin et al. [11] proposed an novel intrusion detection model. This intrusion detection model balances the exploration and exploitation processes by combining the Sine-Cosine algorithm with the Wrapper Modified Whale-Optimization to extract the global optimal features. By using these optimal features, a weighted Extreme Gradient Boost (XGBoost) Classifier with a regularized loss function applied to accurately identify both binary and multiple attacks. Proposed model was evaluated using UNSW-NB15 Multiple attack dataset and CICIDS Binary attack dataset. Performance metrics for UNSW NB15 dataset are as follows; Accuracy: 91%, Precision: 73%, Recall: 90%, and F1 score: 76%, Similarly for CICIDS Binary dataset; Accuracy 0.98, Precision 0.95, Recall 0.98, and F1-score 0.96.

Z. Chen et. al. used XGBoost classifier for DDoS attack detection in software defined network (SDN). In software defined network, controller performs main controlling of the entire network. If controller is attacked whole network becomes inaccessible and denies services to the user. In that case the attack detection becomes crucial at early stage, in order to avoid halt down of the network. In this paper author simulated attack by Hyenae tool on POX as controller. With the help of proposed XGBoost classifier DDoS attack detected with accuracy of 98.53%. Though the reported accuracy was up to the mark, but the suggested model is not robust against outliers or noise [12].

To categorize and forecast common types of attacks Karthick Raghunath et. al. [13] proposed a hybridized classifier consisting XGBoost and Regression classifier. The hybridized classifier—which combines the ideas of the XGBoost and Regression classifiers was implemented in Inception V4, to train and test model additionally. CSE-CIC-IDS2018, UHN Dataset and EMBER dataset were used for evaluation of the proposed model. In-general detection rate recorded during training and testing was 98.18% and 99.13% respectively, but False alarm rate about 2.316%.

The preprocessed web request log data used and proposed a model to identify online threats by Eunaicy et al in [14]. Log files are used to determine the state of an attack. Web threats are detected using Artificial Neural Network (ANN), Convolutional Neural Network (CNN), and Recurrent Neural Network (RNN) algorithms on log data. The authors had a 94% accuracy rate in identifying XSS and SQL injection threats. Z. Tian et al. proposed a system for detecting web attacks by employing a deep learning technique to analyze URLs and distinguish between malicious and legitimate web requests [15]. Web server security systems designed using ML algorithms analyzed and performances evaluated in [16].

These studies stimulate the design of a super learner model to detect attack on web server with improvements in accuracy.

3. Proposed Methodoligy

The proposed methodology for web server attack detection uses super learner ensemble models. Flow of proposed methodology depicted in figure 1. If there are null values or other special characters in the dataset, remove them during the pre-processing and cleaning stage.

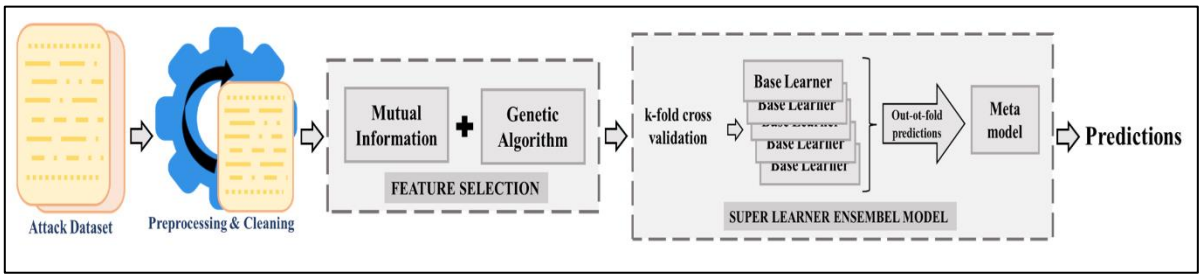

Figure 1. Proposed Methodology for web server attack detection

3.1. Feature Selection

In order to minimize the number of input variables, feature selection techniques are utilized to eliminate irrelevant or expendable variables and decrease the set of features to those that are most pertinent to the machine learning model[17-20] . In this research work feature selection is done using a novel hybrid feature selection method proposed in [21]. This feature selection is a wrapper-based approach where features are ranked and extracted using mutual information scores which are above threshold. Then the genetic algorithm is applied on the extracted feature to select only the relevant features for super learner ensemble algorithm.

3.2. Super Learner Ensemble

An ensemble machine learning approach known as the "super learner" employs several machine learning models to explore a predictive modeling problem and produces predictions that are either as good as or better than those produced by any one model. The super learner approach uses stacked generalization, also known as stacking or blending, in conjunction with k-fold cross-validation. This involves using the same k-folds of the data for prediction across many machine learning models known as base learner and fitting a meta-model to the out-of-fold predictions made by each model. In a provably asymptotic manner, the Super Learner method determines the best combination of base learner fits.

The super learning algorithm, which uses a forward resampling technique, is based mostly on cross validation. A set of observations is randomly divided into k folds, one of which is used as a validation set and the model is then trained by the remaining k − 1 folds. This process is repeated for all k fold until each one is used as validation and remaining for training. The issue of having multiple algorithms and not knowing which to employ is resolved by cross-validation, which also helps us prevent overfitting. The derivation and proof of significant properties of the super learner can be found in van der Laan [22]; the same notations and procedures are used here.

Suppose a dataset S consist of n samples S =(Xi,Yi), were Xi is a set of features and Yi is a target variable dependent on Xi and i = 1,…,n. Aim is to estimate the function $E(Y \mid Xi)$ so that minimize expected loss as in (1)

$$\Phi_0 = \arg min_{\Phi} E[L(Xi, \Phi(Xi))] \quad (1)$$

where L is a loss function often given as the squared error loss L_2: $(Y - \Phi(Xi))^2$.

For attack prediction problems several ML models can be used. Let $\mathscr{L}$ be the set of ML models for attack prediction with cardinality N (no of models in set).

Steps are followed by the super learner as;

i. Apply each ML model on the entire training dataset to estimate $\widehat{\Phi}_m(Xi), m = 1, \dots N$.

ii. Split the dataset S using k-fold cross validation into a training and validation samples of K equal sized folds. Let k be the validation fold and (k-1) folds are training, k = 1,…K. Therefore, let V(k) validation set and T(k) training set of folds.

iii. For k[th] fold train each model in models set $\mathscr{L}$ on T(k) and validate same on V(k), save the output as $\widehat{\Phi}_{m,T(\mathbf{k})}(Xi),\ S \in$ V(k)

iv. Save the outputs of each model in a matrix of size $n\ X\ N$ given as in (2)

$$z = \left\{\widehat{\Phi}_{m,T(\mathbf{v})}\left(X_{V(\mathbf{k})}\right),\ k = 1, \dots K \text{ and } m = 1 \dots N\right\} \quad (2)$$

where $X_{\mathcal{V}(\mathbf{k})} = (X_i : S \in \mathcal{V}(\mathrm{k}))$ input features vector of validation set V(k).

v. A weighted vector α is used to index a group of weighted combinations of the candidate estimators (3).

$$O(z|\alpha) = \sum_{m=1}^{N} \alpha_m \widehat{\Phi}_{k,T(\mathbf{v})}\left(X_{V(\mathbf{v})}\right), \alpha_m \geq 0\ \forall m, \sum_{m=1}^{N} \alpha_m = 1 \quad (3)$$

vi. Estimate weighed vector α which minimizes the cross-validated risk of candidate estimator $\sum_{m=1}^{N} \alpha_m \widehat{\Phi}_m$ for all valid α-combinations (4).

$$\hat{\alpha} = arg \min \sum_{i=1}^{n} (\hat{Y}_i - m(z_i|\alpha))^2 \quad (4)$$

vii. Store $\hat{\alpha}$ with $\widehat{\Phi}_m(W),\ m = 1, 2, \dots, N$ according to the group $O(z|\alpha)$ of weighted combinations to generate eventual super learner model (5).

$$\widehat{\Phi}_{SL}(Xi) = \sum_{m=1}^{N} \hat{\alpha}_m \widehat{\Phi}_m(Xi) \quad (5)$$

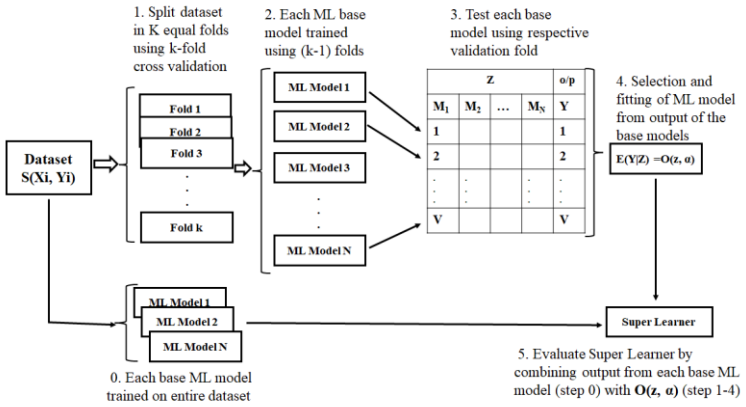

Figure 2. Super Learner Algorithm dataflow

4. Results and Discussion

4.1. Experimental Setup

The study was conducted on a 64-bit Microsoft Windows system running in a VMware Virtual Machine utilizing Python, pandas, and sklearn packages. The virtual machine is set up with two processors and 32GB of RAM, and it is running on a robust Dell Poweredge T440 chassis with an Intel Xeon 4210 CPU running at 2.2GHz, 10C/20T, and 64GB of RAM. This study used UNSW-NB 15 dataset has 49 features of which two dependent variables (target) to identify type of data packets: normal and abnormal (attack). After preprocessing and cleaning the dataset 6,90,215 samples are left. The dataset was imbalances (671638 normal and 18577 attack samples) balanced by oversampling it. Adaptive synthetic sampling (ADASYN) is used in this research work. Adaptive Synthetic Sampling is a technique that creates synthetic samples for minority classes in order to handle imbalanced datasets. By minimizing the bias towards the majority class and balancing the dataset, this oversampling technique enhances classification performance [23].

4.2. Feature Selection

A novel hybrid feature selection method [21] used to select important and relevant features. UNSW-NB 15 dataset used has 47 input features and out of which only six are selected for further processing.

4.3. Base model and super learner model

This research study uses nine base ML models to create super learner algorithm like Ada Boost, Bagging, Decision Tree (DT), Extra Tree, k-NN, Logistic Regression (LR), NB, RF and SVM classifiers. LR is used as super learner meta model to predict the final output.

4.4. Results

The k-fold cross validation method applied on complete dataset (k=10). Out of 10 folds 9 folds are used for training each base ML models and respective 10th fold used for testing. The accuracies of the base models are given in the Table 1 below.

Table 1. Accuracy of base ML models in super learner algorithm

ML Model	Accuracy (%)
Ada Boost	98.13
Bagging	98.19
Decision tree	97.95
Extra tree classifier	98.15
kNN	97.84
Logistic Regression	84.09
Naïve Bayes	97.86
Random forest	98.20
SVM	98.14

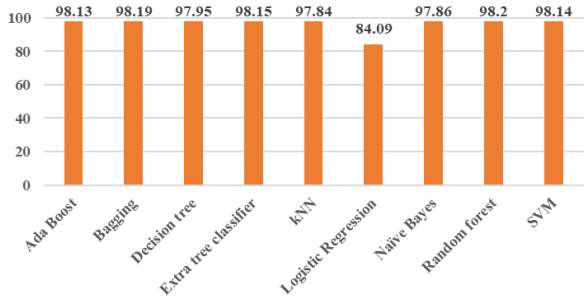

Figure 3. Super Learner Algorithm dataflow

The detection accuracy of each of the base model represented graphically in the Figure. 3. From the figure it clearly shows that the accuracy of the base models are varied from 84% to 98%. With these predictions the meta model is trained and tested. After training of the super learner meta model from the prediction matrix of the base models and the training dataset, the meta model is tested using testing dataset. The performances evaluation metrics of the super learner is given in the Table 2 below

Table 2. Performance metrics of super learner algorithm

ML Model	Accuracy (%)
Accuracy	99.69
Precision	99.45
Recall	99.98
F1-score	99.70
FPR	0.58
FNR	3.10

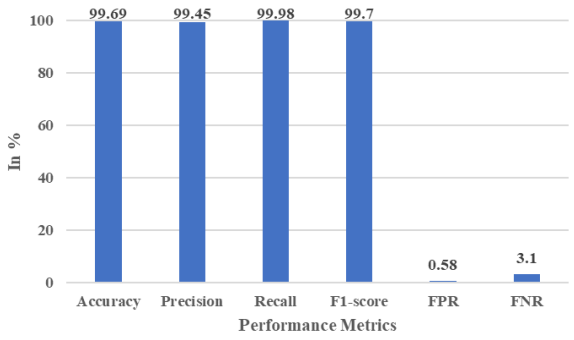

Figure 4. Super Learner Algorithm dataflow

In this study Logistic regression algorithm is used as meta model which is trained and tested on the predictions of the base models. LR gives best performance as a super learner algorithm, compared to the other base learner algorithms. During the base learning phase the accuracy of the LR is quiet low compared the other base learners. In the super learning phase it performs exceptionally good with detection accuracy, precision, recall and f1score above 99%. Also the false positive and false negative rates are less.

The performance metrics are also evaluated for the KDD CUP 99 data set using proposed super learner algorithm. The accuracy of base models in super learner are given in the Table 3 and the accuracy of meta algorithm with other performance parameters are given in Table 3.

Table 3. Accuracy of base ML models in super learner algorithm for KDD CUP99 dataset

ML Model	Accuracy (%)
Ada Boost	97.80
Bagging	96.80
Decision tree	96.85
Extra tree classifier	96.40
kNN	96.80
Logistic Regression	95.87
Naïve Bayes	64.67
Random forest	97.80
SVM	60.46

Table 4. Performance metrics of super learner algorithm

ML Model	Accuracy (%)
Accuracy	99.90
Precision	99.93
Recall	99.79
F1-score	99.86
FPR	0.064
FNR	6.087

The accuracy values in the Table 6 show that the NB and SVM predicts poor detection for KDD CUP 99 datasets but the overall super learner performance is exceptionally well.

4.5. Comparative Analysis

The obtained results are compared with the results reported in the existing literature using UNSW NB 15 datasets (Table 5).

Table 5. Comparison of accuracy obtained and reported in the literature

	Method used	Accuracy (%)
G. Mohiuddin et al [11]	XGBoost	91.00
A. D. Vibhute et al [24]	CNN	99.00
Kasongo et al [25]	XGBoost with kNN	95.86
Ahmad M. et al [26]	SVM	97.69
Proposed	**Super Learner**	**99.69**

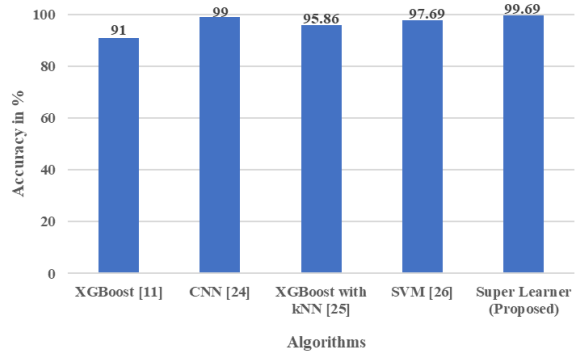

Figure 5. Super Learner Algorithm dataflow

The comparative analysis of result obtained and the results reported in literature[11], [24], [25] and [26] is shown in the figure 5. It clearly shows that the proposed method of attack detection outperforms the state of art and the detection accuracy improved up to 9.54%.

5. Conclusion

In this study, a super learner ensemble learning model was proposed and evaluated to improve web server security by effectively detecting attacks. Through combining predictions from multiple machine learning models, the super learner approach demonstrates a superior ability to classify and detect web attacks compared to individual models. The ensemble model successfully addresses challenges like high false-positive rates and imbalanced data by leveraging adaptive synthetic sampling for data balancing and feature selection for optimization. The experimental results validate the efficiency of the super learner model, achieving a remarkable detection accuracy of 99.69% on the UNSW-NB 15 dataset and 99.90% on the KDD CUP 99 dataset, outperforming traditional methods. Additionally, the model achieved robust precision, recall, and F1-scores, showcasing high reliability in distinguishing between normal and malicious traffic. Comparative analysis further highlights that the super learner algorithm surpasses existing approaches, offering a significant improvement in accuracy.

This research illustrates that ensemble learning methods, particularly the super learner model, can be powerful tools in cybersecurity applications. Future work may explore different combinations of base models, alternative datasets, or the integration of real-time anomaly detection mechanisms to enhance the applicability of this model in diverse web environments.

References

[1] A. Tekerek, "A novel architecture for web-based attack detection using convolutional neural network,," Computers & Security, vol. 100, p. 102096, 2021.
[2] "Enisa Threat Landscape Report," European Union Agency for Cybersecurity (ENISA), 2023.
[3] Patil S., Vanmali A., Bansode R., "Cyber Security Concerns for IoB," in Internet of Behaviors (IoB), CRC Press, 2023, p. 141–155.
[4] S. H. Najla Odeh, "Detecting and Preventing Common Web Application Vulnerabilities: A Comprehensive," International Journal of Information Technology and Computer Science(IJITCS), vol. 15, no. 3, pp. 26-41, 2023.
[5] P. V. V. T. U. P. Mishra, "detailed investigation and analysis of using," IEEE, vol. 21, p. 686–728, 2019.
[6] J. V. and P. Casas, "Ensemble-learning Approaches for Network Security and Anomaly Detection," in In Proceedings of the Workshop on Big Data Analytics and Machine Learning for Data Communication Networks (Big-DAMA '17), Los Angeles CA USA, 2017.

[7] S. V. K. S. A. T. T. S. K. T. G. K. K.A. Dhanya, "Detection of Network Attacks using Machine Learning and Deep Learning Models," in International Conference on Machine Learning and Data Engineering, 2023.
[8] S. M. Dong S, "DDoS Attack Detection Method Based on Improved KNN With the Degree of DDoS Attack in Software-Defined Networks," IEEE Access, vol. 8, p. 5039–5048, 2020.
[9] H. B. D. M. Sharif, "Detection of application-layer DDoS attacks using machine learning and genetic algorithms," Computers & Security, vol. 135, p. 14, 2023.
[10] S. Ismail, Z. El Mrabet and H. Reza, "An Ensemble-Based Machine Learning Approach for Cyber-Attacks Detection in Wireless Sensor Networks," Applied Sciences, vol. 13, p. 30, 2023.
[11] L. Z. Z. J. J. W. W. L. F. W. S. C. J. Z. X. G. Mohiuddin, "Intrusion Detection using hybridized Meta-heuristic techniques with Weighted XGBoost Classifier," Expert Systems With Applications, vol. 232, p. 120596, 2023.
[12] F. J. Y. C. X. G. W. L. a. J. P. Z. Chen, "GBoost Classifier for DDoS Attack Detection and Analysis in SDN-based Cloud," 2018 IEEE International Conference on Big Data and Smart Computing, pp. 251-256, 2018.
[13] V. V. K. M. V. K. K. S. T. R. M. a. A. S. K. M. K. Raghunath, "XGBoost Regression Classifier (XRC) Model for Cyber Attack Detection and Classification Using Inception V4," Journal of Web Engineering, vol. 21, no. 04, p. 1295–1322, 2022.
[14] C. S. S. Eunaicy, "Web attack detection using deep learning models," Materials Today: Proceedings, vol. 62, pp. 4806-4813, 2022.
[15] C. L. J. Q. X. D. a. M. G. Z. Tian, "A Distributed Deep Learning System for Web Attack Detection on Edge Devices," IEEE Transactions on Industrial Informatics, vol. 16, no. 3, pp. 1963-1971, 2020.
[16] S. P. a. R. Bansode, "Performance Evaluation of Web Server Security Systems Designed using Machine Learning Approach," Mukt Shabd Journal, vol. 3, no. 4, 2024.
[17] F. C. S. K. A. N. H. I. A. H. Azmi MAH, "Feature Selection Approach to Detect DDoS Attack Using Machine Learning," JOIV : International Journal on Informatics Visualization, vol. 5, no. 4, p. 395–401, 2021.
[18] S. K. C. Y. Chanu US, "A dynamic feature selection technique to detect DDoS attack," Journal of Information Security and Applications, vol. 74, no. C, p. 14, 2023.
[19] S. K. Deepak Kshirsagar, "An efficient feature reduction method for the detection of DoS attack," ICT Express, vol. 7, no. 3, pp. 371-375, 2021.
[20] S. H. Y. W. Jingyi Su, "Features selection and prediction for IoT attacks," High-Confidence Computing, vol. 2, no. 2, pp. 1-6, 2022.
[21] B. R. Patil S, "A Hybrid Feature Selection Approach Incorporating Mutual Information and Genetics Algorithm for Web Server Attack Detection," Indian Journal of Science and Technology, vol. 17, no. 4, pp. 325-332, 2024.
[22] P. E. H. A. van der Laan MJ, "Super Learner," Stat Appl Genet Mol Biol, vol. 6, no. 1, 2007.
[23] Y. G. F. H. Jingmei Liu, "A fast network intrusion detection system using adaptive synthetic oversampling and LightGBM," Computers & Security, vol. 106, 2021.
[24] M. K. C. H. P. S. V. G. A. V. M. K. K. P. A. D. Vibhute, "Network anomaly detection and performance evaluation of Convolutional Neural Networks on UNSW-NB15 dataset," Procedia Computer Science, vol. 235, pp. 2227-2236, 2024.
[25] S. S. Y. Kasongo, "Performance Analysis of Intrusion Detection Systems Using a Feature Selection Method on the UNSW-NB15 Dataset," Journal of Big Data, vol. 7, 2020.
[26] M. R. Q. Z. M. e. a. Ahmad, "Intrusion detection in internet of things using supervised machine learning based on application and transport layer features using UNSW-NB15 data-set," EURASIP Journal on Wireless Communications and Networking, vol. 10, 2021.