JAIEP

Anticipating Malicious Server Attacks: Evaluating the Effectiveness of Various Machine Learning Models

Al-Seyday.T. Qenawy [1] *, **Muhammad Ahsan**[5]
[1]Intelligent Systems and Machine Learning Lab, Shenzhen 518000, China
[2]School of Mathematical Sciences, Jiangsu University, Jiangsu 212013, China

*Corresponding author: S.Qenawy@asia.com
Emails: S.Qenawy@asia.com, ahsan1826@gmail.com

Abstract

The global shift to online payments means that companies face growing cyber dangers, especially to servers. The target of this analysis is on malicious server hacks to be forecasted based on anonymized incident data of several features that are logging parameters and an outcome variable of hack occurrence. Based on the problems context, several machine learning models were created and tested, such as K-Nearest Neighbors, Na¨ıve Bayes, Neural Networks, Gradient Boosting, and finally, the e SVM with the RBF Kernel for the prediction of possible server hacks. The models were evaluated according to the performance indicators such as accuracy, sensitivity, specificity, precision, and F1 measure. As for the models, the highest accuracy was recorded for K-Nearest Neighbors with 93.5% while still revealing the highest sensitivity, making it the best model in making a prognosis on server hacks. The second model, the Neural Network, also demonstrated good results regarding Sensitivity and F1-score. Based on our study, it is evident that these machine learning models can be used to predict possible future server hacks thus acting as a preventive measure in cybersecurity. This paper has explored the practical application of machine learning in cybersecurity and other related topics, while future work is expected to look at other advanced models and other features that would improve the recognition's accuracy.

Keywords: Machine Learning, Cybersecurity, Server Hack Prediction, Anonymized Data, Comparative Analysis
MSC: 68T07; 62J05; 93B45

Doi: https://doi.org/10.21608/jaiep.2024.314522.1007
Received: August 22, 2024 Revised: October 20, 2024 Accepted: November 27, 2024

1 Introduction

In the realm of cybersecurity, classifying server hacks as a category of malware is crucial due to the increasing prevalence of cyber-attacks. Servers, which play a vital role in handling and storing information within communication networks, are prime targets for attackers seeking system loopholes to gain unauthorized access, steal information, or deny services to legitimate users. These escalating threats necessitate advanced classification methods for various server hacks [1–3].

Malicious server hacks encompass several types, including SQL injection, cross-site scripting (XSS), denial-of-service (DoS) attacks, and Advanced Persistent Threats (APTs). Each attack type possesses distinct characteristics and behavioral patterns, which are pivotal for classification. Classification in this context refers to employing machine learning and data analysis to categorize attacks based on attributes such as attack methods, exploitation techniques, and indicators of compromise (IoCs) [4–6].

This paper aims to demonstrate how classification techniques can be employed to identify malicious server hacks. The study proposes the use of machine learning algorithms such as decision trees, random forests, and neural networks to analyze a dataset containing features like attack protocols, flow patterns, and interaction logs. The primary objectives include building an accurate classifier for encoding various server hacks and identifying key features that differentiate attack types. For example, the model might classify attacks into SQL injection, XSS, and DoS, based on how each manifests in server logs and network activity [7–9].

Furthermore, the findings of this research are expected to enhance server security and incident management. Accurate classification of malicious hacks enables organizations to promptly identify and counteract attacks. For instance, if a classification model detects an SQL injection pattern, automated measures could block the attack or alert the security team. Additionally, understanding the characteristics of different attacks contributes to better security measures, preventing future incidents [10–12].

Thus, integrating classification techniques into the detection of malicious server hacks is vital for strengthening cybersecurity measures and safeguarding critical server frameworks. This paper seeks to contribute to the current body of literature by categorizing various attacks and providing recommendations for improving organizational network security. The overarching goal is to lay the groundwork for more robust security systems capable of countering the ever-evolving threat landscape.

2 Literature Review

The classification of malicious server hacks is a pivotal domain in cybersecurity, aimed at identifying various types of attacks targeting servers. Effective classification aids in understanding threats, improving defenses, and mitigating risks.

Early classifiers, such as decision trees and random forests, have been widely used to categorize server hacks [13]. Decision trees are intuitive tools that segregate attacks based on attributes like attack type, exploited vulnerabilities, and target system characteristics. Random forests, as ensemble learning methods, combine multiple decision trees to improve classification accuracy and manage data correlation issues [14].

Support Vector Machines (SVMs) have also been employed for their efficiency in handling high-dimensional spaces and non-linear interactions [15]. SVMs can classify attacks such as SQL injection, XSS, and buffer overflow by finding optimal hyperplanes that separate attack classes based on features like payload characteristics and attack patterns. Deep learning models, including Neural Networks and Convolutional Neural Networks (CNNs), have shown promise in analyzing server hack data [16]. These models learn high-level features from raw data, such as network traffic logs and system call sequences, achieving high accuracy in classifying malicious activities.

Anomaly detection techniques are another critical approach to identifying suspicious activity indicative of illegal server hacking [17]. Techniques like isolation forests and autoencoders detect abnormal patterns in server access and usage. Intrusion Detection Systems (IDS) leverage classification algorithms to monitor server activity in real-time, identifying potential malicious actions [18]. Machine learning-enhanced IDS adapts to emerging attack patterns, improving detection capabilities.

Feature engineering and selection play a significant role in optimizing classification models for identifying malicious server hacks [19]. Methods like Principal Component Analysis (PCA) and Recursive Feature Elimination (RFE) help identify the most relevant features, while feature construction from raw data, such as attack time and payload size, enhances model performance. Combining classification techniques, such as STACKING and BOOSTING, has been proposed to improve accuracy and reliability [20].

Real-time classification solutions, enabled by stream processing and distributed machine learning frameworks, are essential for live environments [21]. These approaches analyze big data to identify malicious activities in real-time, preventing or mitigating server attacks. Examining real-life scenarios and applying corresponding

Doi: https://doi.org/10.21608/jaiep.2024.314522.1007
Received: August 22, 2024 Revised: October 20, 2024 Accepted: November 27, 2024

classification methods reveal insights into combating server hacks [22]. Such analyses highlight the effectiveness of various methods and provide practical guidance for enhancing cybersecurity measures.

3 Dataset Description

3.1 Dataset Overview

The dataset used for this study is a binary classification dataset aimed at determining the likelihood of a server being hacked, given several anonymized variables. Research has revealed that it contains numerous types of incident logs characterized by the `INCIDENT ID`. All logs are pre-fixed with the date (`DATE`), providing a temporal reference to each incident. The dataset includes fifteen logging parameters anonymized and numbered as `X1` to `X15`. The target variable, `MALICIOUS OFFENSE`, is binary and equals 1 if the specific incident stems from a server hack (i.e., `Yes`) and 0 otherwise (`No`).

The primary objective of the study is to develop models capable of forecasting the target variable based on these anonymized parameters.

3.2 Data Preprocessing

The dataset was subjected to several preprocessing steps before being fed into the models, as detailed below:

Handling Missing Data Missing data in the dataset was addressed as a preliminary step. Missing values, depending on their nature and distribution, were either replaced using the arithmetic mean/median of the respective feature or removed if their presence was significant and non-random. This approach ensures that the effectiveness of machine learning algorithms is not compromised.

Anonymization To prevent data leakage and maintain privacy, the data was anonymized. This step ensured that no individual or organizational identity could be inferred from the dataset while retaining its analytical utility.

Normalization The features `X1` to `X15` were normalized to bring their values within a standard range, typically [0,1]. This process is crucial for machine learning models, especially distance-based models like K-Nearest Neighbors (KNN), as it prevents features with large magnitudes from dominating the learning process.

Feature Engineering Feature engineering was conducted to improve model accuracy by creating new features or modifying existing ones. Techniques included polynomial feature creation and constructing interaction terms (e.g., means, sums, or ratios of raw variables). These engineered features were included based on dependencies and correlations identified during exploratory data analysis.

Visualization and Correlation Analysis

- **Andrew Curves:** Figure 1 illustrates Andrew curves for the dataset, providing a visualization of high-dimensional data. Each curve represents an instance, revealing potential clusters, outliers, and patterns within the dataset.

- **Correlation Matrix:** Figure 2 shows a heatmap of the correlation matrix. Rows and columns represent the dataset's features, and each cell displays the correlation coefficient. The intensity of the color indicates the degree of correlation, aiding in feature selection, dimensionality reduction, and addressing multicollinearity issues.

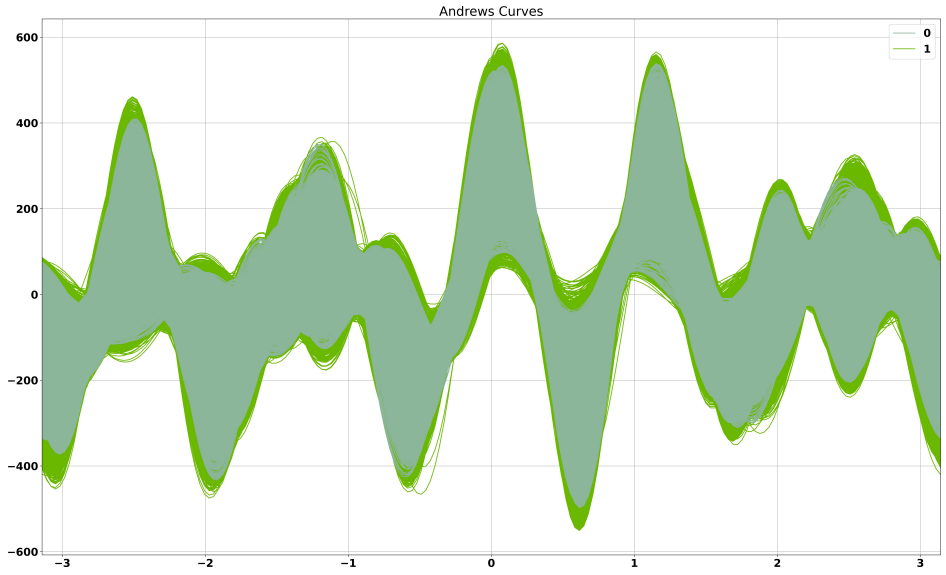

Figure 1: Andrew Curves for the dataset

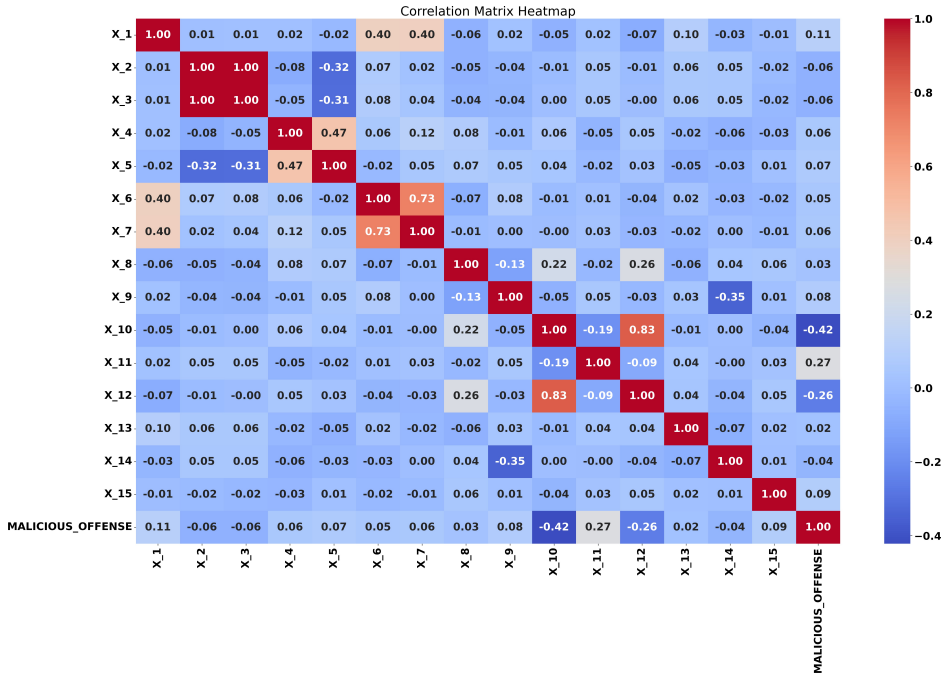

Figure 2: Heatmap of Correlation Matrix for the dataset

Data Splitting and Validation The dataset was split into training and testing sets, with 70-80% allocated for training and 20-30% for testing. This ensures that the models are evaluated on unseen data, replicating real-world scenarios. Additionally, k-fold cross-validation ($k = 10$) was applied to the training set to optimize hyperparameters, reduce variance, and prevent overfitting.

4 Methodology

4.1 Machine Learning Models

This study utilized various machine learning models with distinct characteristics to predict server hacks using anonymized incident data. The models employed are described below:

Doi: https://doi.org/10.21608/jaiep.2024.314522.1007
Received: August 22, 2024 Revised: October 20, 2024 Accepted: November 27, 2024

- **K-Nearest Neighbors (KNN):** KNN assigns an instance to the class most represented among its nearest neighbors in the feature space. It is non-parametric, meaning it does not require data to be normally distributed. This method is effective when local structure is strong, as similar instances are expected to yield similar outcomes.
- **Naive Bayes:** Based on Bayes' theorem, Naive Bayes assumes independence among features. Despite this assumption being 'naive,' it performs well in high-dimensional problems. It is computationally efficient and provides excellent results when feature independence is approximately valid.
- **Neural Network (Multi-Layer Perceptron, MLP):** MLP is a type of Artificial Neural Network (ANN) comprising multiple layers of neurons. It models nonlinear relationships between inputs and outputs. Typical components include an input layer, one or more hidden layers, and an output layer. MLP is particularly suited to complex dependencies.
- **Gradient Boosting:** An ensemble learning technique where models are sequentially built to correct errors of previous models. Gradient Boosting is highly effective in classification tasks, yielding accurate models with reduced overfitting.
- **Support Vector Machine (SVM) with RBF Kernel:** SVM identifies the optimal hyperplane to separate classes in the feature space. Using the RBF kernel, SVM handles nonlinear relationships effectively, making it suitable for high-dimensional data.

4.2 Model Training

The training process for each model included steps to optimize performance and ensure robust predictions:

- **K-Nearest Neighbors (KNN):** The primary parameter, k (number of neighbors), was selected using k-fold cross-validation and grid search to balance bias and variance. The distance metric (e.g., Euclidean or Manhattan) was also optimized.
- **Naive Bayes:** With fewer hyperparameters, Naive Bayes focused on meeting model assumptions. Gaussian Naive Bayes was used, with tuning for priors and smoothing parameters to handle zero-probability issues.
- **Neural Network (MLP):** Key parameters included the number of hidden layers, neurons per layer, learning rate, and activation functions. Grid search and cross-validation determined the best configuration. Dropout and early stopping were applied to prevent overfitting.
- **Gradient Boosting:** Parameters such as the number of boosting rounds, learning rate, and tree depth were tuned. Grid search and cross-validation were employed to optimize performance and prevent overfitting.
- **SVM with RBF Kernel:** Parameters C (margin error tradeoff) and γ (influence of training examples) were optimized using grid search with cross-validation for the best fit.

4.3 Evaluation Metrics

The models were evaluated using the following metrics to ensure comprehensive performance assessment:

- **Accuracy:** The ratio of correctly predicted instances to the total number of predictions. While a general measure of performance, it can be misleading for imbalanced datasets.
- **Sensitivity (True Positive Rate):** The proportion of actual positives correctly identified. This metric is critical for detecting hacks and avoiding false negatives.
- **Specificity (True Negative Rate):** The proportion of actual negatives correctly identified. It ensures the model minimizes false alarms.
- **Precision (Positive Predictive Value):** The ratio of true positives to total predicted positives. High precision ensures the model accurately isolates hacks.
- **Negative Predictive Value (NPV):** The ratio of true negatives to total predicted negatives. NPV ensures reliable predictions when the model forecasts no hack.

- **F1-Score:** The harmonic mean of precision and recall, balancing these metrics. It is especially useful for imbalanced datasets, accounting for both false positives and false negatives.

These metrics were selected for their ability to objectively evaluate the models' performance, particularly in predicting malicious server attacks. Sensitivity, specificity, and the F1-score were emphasized due to their importance in minimizing adverse consequences associated with false positives and negatives.

5 Results

The analysis comparing the performance of various machine learning models in predicting malicious server hacks is summarized below. The evaluation metrics include Accuracy, Sensitivity, Specificity, Precision, Negative Predictive Value, and F1-Score. These metrics provide a comprehensive assessment of each model's ability to identify real hacks (sensitivity), avoid false alarms (specificity), and achieve reasonable accuracy.

The results reveal the strengths and weaknesses of each model, offering insights into their practical applicability in combating cyber threats. Table 1 presents the performance metrics of the five machine learning models evaluated in this study.

Table 1: Performance Metrics of Various Machine Learning Models

Models	Accuracy	Sensitivity (TRP)	Specificity (TNP)	Precision (PPV)	NPV	F1-Score
K-Nearest Neighbors	0.935	0.9604	0.9091	0.9151	0.9574	0.9372
Naive Bayes	0.915	0.9307	0.8990	0.9038	0.9271	0.9171
Neural Network (MLP)	0.910	0.9406	0.8788	0.8879	0.9355	0.9135
Gradient Boosting	0.890	0.8614	0.9192	0.9158	0.8667	0.8878
SVM (RBF Kernel)	0.885	0.8713	0.8990	0.8980	0.8725	0.8844

The KNN model demonstrated a sensitivity of 96.04% and an overall accuracy of 93.5%, emerging as the best-performing model in this study. This highlights its efficiency in capturing local data structures, which is crucial in classification problems such as distinguishing between hacks and non-hacks. Its high F1-Score further indicates an optimal trade-off between precision and recall, crucial in operations where both false positives and false negatives have significant implications.

Figures 3 and 4 visualize the performance results of the models. Figure 3 presents box plots for the distribution of metrics such as accuracy, precision, and recall, offering insights into their variability, median values, and potential outliers. Figure 4 depicts the sensitivity (True Positive Rate) of the models, allowing direct comparison of their effectiveness in identifying positive instances.

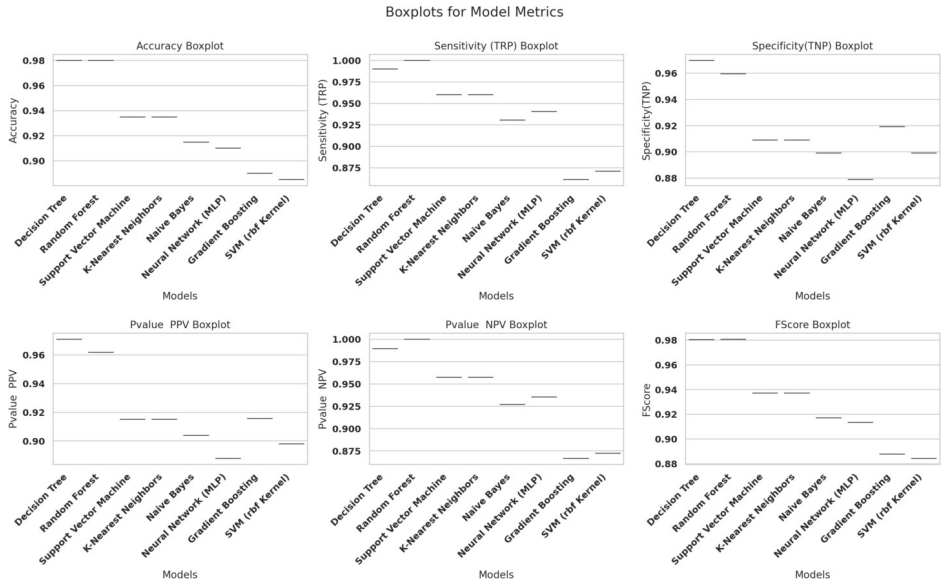

Figure 3: Box Plots of the Machine Learning Model Performance

Doi: https://doi.org/10.21608/jaiep.2024.314522.1007
Received: August 22, 2024 Revised: October 20, 2024 Accepted: November 27, 2024

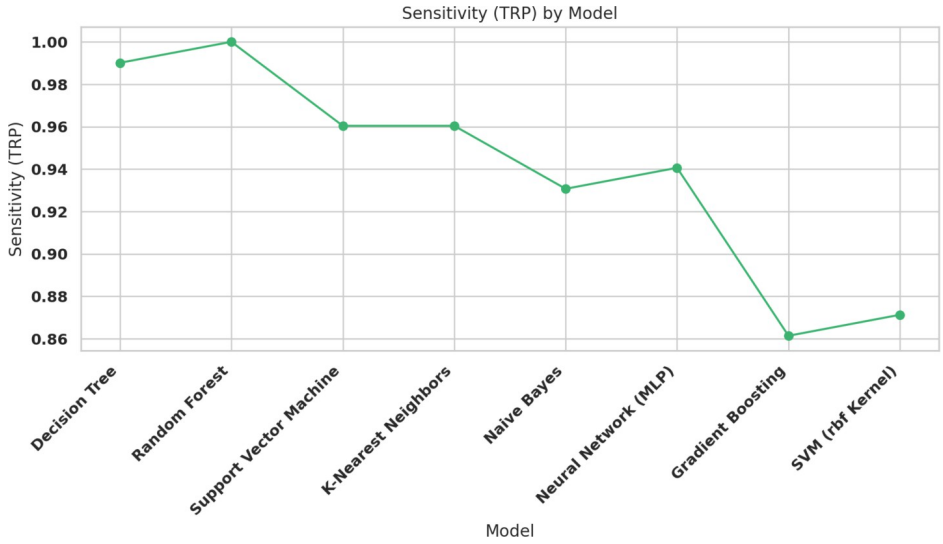

Figure 4: Sensitivity by Machine Learning Models

The findings suggest that KNN is the most suitable model for this dataset, balancing sensitivity, specificity, and overall accuracy effectively. However, the choice of the model should also consider the context of application, including tolerable levels of false positives and false negatives in cybersecurity scenarios.

6 Conclusion

This study evaluated five machine learning models—K-Nearest Neighbors, Naive Bayes, Neural Network (MLP), Gradient Boosting, and Support Vector Machine with RBF Kernel—on anonymized incident data to predict malicious server hacks. KNN emerged as the most effective model, achieving an accuracy of 93.5%, an F1-Score of 0.9372, and a sensitivity of 96.04%.

These findings highlight the utility of KNN and similar models in cybersecurity, helping organizations anticipate and prevent server hacks. However, the study had limitations, such as the use of anonymized data and the potential for data bias, which may affect model generalization.

Future work should focus on developing more sophisticated models, utilizing detailed and dependent data, and improving model interpretability for practical applications. Testing and refining these models in real-world environments with ongoing updates can further enhance their effectiveness in countering emerging cyber threats and ensuring robust information security across various domains.

References

[1] A. Alshammari and A. Aldribi. Apply machine learning techniques to detect malicious network traffic in cloud computing. *Journal of Big Data*, 8(1):90, 2021.

[2] M. Amanowicz and D. Jankowski. Detection and classification of malicious flows in software-defined networks using data mining techniques. *Sensors*, 21(9), 2021.

[3] M. Arunkumar and K. Ashok Kumar. Malicious attack detection approach in cloud computing using machine learning techniques. *Soft Computing*, 26(23):13097–13107, 2022.

[4] M. Arunkumar and K. A. Kumar. Gosvm: Gannet optimization based support vector machine for malicious attack detection in cloud environment. *International Journal of Information Technology*, 15(3):1653–1660, 2023.

[5] B. Biswas, A. Mukhopadhyay, S. Bhattacharjee, A. Kumar, and D. Delen. A text-mining based cyber-risk assessment and mitigation framework for critical analysis of online hacker forums. *Decision Support Systems*, 152:113651, 2022.

[6] S. Chng, H. Y. Lu, A. Kumar, and D. Yau. Hacker types, motivations and strategies: A comprehensive framework. *Computers in Human Behavior Reports*, 5:100167, 2022.

[7] Z. Ismail, A. Jantan, Mohd. N. Yusoff, and M. U. Kiru. The effects of feature selection on the classification of encrypted botnet. *Journal of Computer Virology and Hacking Techniques*, 17(1):61–74, 2021.

[8] S. Kaddoura. Classification of malicious and benign websites by network features using supervised machine learning algorithms. In *2021 5th Cyber Security in Networking Conference (CSNet)*, pages 36–40, 2021.

[9] Ö. Kasim. An ensemble classification-based approach to detect attack level of sql injections. *Journal of Information Security and Applications*, 59:102852, 2021.

[10] R. Komatwar and M. Kokare. Retracted article: A survey on malware detection and classification. *Journal of Applied Security Research*, 2021.

[11] K. Lee, J. Lee, and K. Yim. Classification and analysis of malicious code detection techniques based on the apt attack. *Applied Sciences*, 13(5), 2023.

[12] S. Li, Y. Li, W. Han, X. Du, M. Guizani, and Z. Tian. Malicious mining code detection based on ensemble learning in cloud computing environment. *Simulation Modelling Practice and Theory*, 113:102391, 2021.

[13] A. Mallik, A. Khetarpal, and S. Kumar. Conrec: Malware classification using convolutional recurrence. *Journal of Computer Virology and Hacking Techniques*, 18(4):297–313, 2022.

[14] P. Maniriho, A. N. Mahmood, and M. J. M. Chowdhury. A study on malicious software behaviour analysis and detection techniques: Taxonomy, current trends and challenges. *Future Generation Computer Systems*, 130:1–18, 2022.

[15] G. Palaniappan, S. S, B. Rajendran, S. Sanjay, Goyal, and B. B S. Malicious domain detection using machine learning on domain name features, host-based features and web-based features. In *Procedia Computer Science*, volume 171, pages 654–661, 2020.

[16] M. Rabbani, Y. L. Wang, R. Khoshkangini, H. Jelodar, R. Zhao, and P. Hu. A hybrid machine learning approach for malicious behaviour detection and recognition in cloud computing. *Journal of Network and Computer Applications*, 151:102507, 2020.

[17] S. K. J. Rizvi, W. Aslam, M. Shahzad, S. Saleem, and M. M. Fraz. Proud-mal: Static analysis-based progressive framework for deep unsupervised malware classification of windows portable executable. *Complex & Intelligent Systems*, 8(1):673–685, 2022.

[18] D. Singh, A. Shukla, and M. Sajwan. Deep transfer learning framework for the identification of malicious activities to combat cyberattack. *Future Generation Computer Systems*, 125:687–697, 2021.

[19] S. Srinivasan and P. Deepalakshmi. Enetrm: Elasticnet regression model based malicious cyber-attacks prediction in real-time server. *Measurement: Sensors*, 25:100654, 2023.

[20] A. Tekerek. A novel architecture for web-based attack detection using convolutional neural network. *Computers & Security*, 100:102096, 2021.

[21] K. N. K. Thapa and N. Duraipandian. Malicious traffic classification using long short-term memory (lstm) model. *Wireless Personal Communications*, 119(3):2707–2724, 2021.

[22] N. Usman, S. Usman, F. Khan, M. A. Jan, A. Sajid, M. Alazab, and P. Watters. Intelligent dynamic malware detection using machine learning in ip reputation for forensics data analytics. *Future Generation Computer Systems*, 118:124–141, 2021.

Doi: https://doi.org/10.21608/jaiep.2024.314522.1007
Received: August 22, 2024 Revised: October 20, 2024 Accepted: November 27, 2024