Journal of Information System Research (JOSH)
Volume 6, No. 1, Oktober 2024, pp 619−628
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v6i1.6089

Evaluating the Effectiveness of Machine Learning Models for Cyberattack Detection: A Study on Model Generalization and Dataset Imbalance

Gregorius Airlangga

Engineering Faculty, Information System Study Program, Atma Jaya Catholic University of Indonesia, Jakarta
Jl. Jend. Sudirman No.51 5, RT.004/RW.4, Karet Semanggi, Setiabudi District, South Jakarta City, Special Capital Region of Jakarta, Indonesia
Email: gregorius.airlangga@atmajaya.ac.id
Correspondence Author Email: gregorius.airlangga@atmajaya.ac.id
Submitted: **17/10/2024;** Accepted: **29/10/2024;** Published: **31/10/2024**

Abstract−In today's rapidly evolving digital landscape, detecting and preventing cyberattacks has become crucial for securing networks and data. This study evaluates the performance of several machine learning models, including RandomForest, GradientBoosting, XGBoost, LightGBM, CatBoost, Support Vector Classifier (SVC), Logistic Regression, and an ensemble Voting Classifier, in detecting and classifying cyberattacks. The models were tested on a real-world cybersecurity dataset with significant class imbalance, where benign traffic vastly outnumbers malicious attacks. Results showed that while some models, such as RandomForest and the Voting Classifier, achieved high training accuracy, they suffered from overfitting, with test accuracies not exceeding 34%. Boosting models like XGBoost and LightGBM exhibited better generalization than RandomForest but still struggled to handle the dataset complexity. The primary limitations of this study include the dataset's imbalance, the high dimensionality of the features, and the models' tendency to overfit. These challenges highlight the need for more robust data preprocessing techniques, hyperparameter tuning, and exploration of advanced models, such as deep learning architectures, for future work. The findings provide insights into the challenges of using machine learning for cybersecurity attack detection and point toward future directions for improving model performance in real-world settings.

Keywords: Cyberattack Detection; Machine Learning; Imbalanced Datasets; Model Overfitting; XGBoost; RandomForest

1. INTRODUCTION

The growing reliance on digital technologies has drastically transformed the way organizations operate, enabling enhanced connectivity, efficiency, and service delivery [1]–[3]. However, this transformation has also introduced significant risks, as cyberattacks have escalated in both frequency and complexity. Modern cyber threats, such as ransomware, data breaches, phishing, and Advanced Persistent Threats (APTs), have the potential to cause devastating financial, operational, and reputational damage to organizations [4]. The World Economic Forum ranks cyberattacks among the top global risks, with the average cost of a data breach reaching $4.35 million in 2023 [5].

This increasing threat landscape highlights the urgent need for robust, adaptive cybersecurity measures capable of addressing these challenges [6]. Traditional cybersecurity approaches, such as signature-based detection systems and rule-based defenses, are becoming less effective in protecting against these modern threats [7]. Attackers now employ sophisticated tactics that allow them to bypass static defenses, such as polymorphic malware that continuously changes its characteristics or zero-day exploits that take advantage of unknown vulnerabilities [8].

As a result, there is growing interest in leveraging machine learning (ML) to enhance cybersecurity systems, with ML models offering the ability to analyze large volumes of data, identify patterns, and detect both known and novel attacks in real time [9]. However, despite the potential of machine learning, there is no consensus on which models are best suited for cybersecurity, especially when applied to diverse and complex real-world cyberattack scenarios [10].

The urgency for advanced machine learning-based cybersecurity solutions has never been greater. With the expansion of cloud services, the proliferation of Internet of Things (IoT) devices, and the widespread adoption of remote work, the attack surface for malicious actors has grown significantly [11]. High-profile incidents like the Colonial Pipeline ransomware attack and the SolarWinds supply chain breach have shown the extent of disruption that modern cyberattacks can cause [12].

These incidents not only result in operational shutdowns but also incur significant financial costs for businesses and governments [13]. The rapid pace at which cyber threats evolve further exacerbates the challenge, making it critical to explore dynamic, scalable solutions that can mitigate the impact of attacks in real-time environments [14]. Current research into machine learning for cybersecurity has produced promising results. Machine learning models have been successfully applied to detect various types of cyber threats. For instance, [15] used deep learning models like Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) to detect anomalies in network traffic, achieving impressive accuracy in identifying suspicious activities.

Similarly, [16] explored ensemble learning techniques, including Random Forest and Gradient Boosting, to detect Distributed Denial of Service (DDoS) attacks. Ensemble models, which combine the strengths of multiple

Journal of Information System Research (JOSH)
Volume 6, No. 1, Oktober 2024, pp 619−628
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v6i1.6089

classifiers, have been shown to outperform individual models by providing more robust and accurate threat detection.

Despite these advances, significant gaps remain in the current body of research. One of the major challenges is the issue of dataset imbalance, where benign network traffic far exceeds malicious traffic [17]. This imbalance can lead to biased models that are more likely to misclassify attacks as normal behavior. Techniques such as the Synthetic Minority Over-sampling Technique (SMOTE) have been proposed to address this imbalance by generating synthetic attack data, but questions remain about the generalizability of these models across various attack types [18]. Additionally, the interpretability of many machine learning models poses another challenge. Complex models, such as deep learning algorithms, often act as "black boxes," making it difficult for cybersecurity professionals to understand how decisions are made.

This lack of transparency limits the practical use of machine learning models in critical security environments where explainability is crucial. Thus, while machine learning has demonstrated potential in enhancing cybersecurity, there are still important gaps to address: models must be able to generalize across a variety of attack types, balance accuracy with computational efficiency, and provide interpretability to meet the needs of real-world applications [19]–[21]. These gaps emphasize the need for a comprehensive comparison of machine learning models to identify which models perform best in addressing these challenges [22]–[24].

The main goal of this research is to conduct a comparative evaluation of several machine learning models in detecting and classifying cyberattacks. Specifically, this study focuses on models such as RandomForest, GradientBoosting, XGBoost, LightGBM, CatBoost, and Support Vector Classifier (SVC) using a large, diverse cybersecurity dataset. By systematically comparing the performance of these models, this research seeks to determine which algorithms provide the best trade-off between detection accuracy, computational efficiency, and interpretability. Conducting this comparison is essential because, despite the progress made in ML applications for cybersecurity, there is no clear understanding of which models are most effective across different attack types in real-world scenarios.

This study contributes to the field by offering a detailed analysis of multiple machine learning models applied to a realistic cybersecurity dataset. Unlike previous studies that often focus on specific attack types or isolated metrics, this research evaluates the performance of models across a wide range of attack signatures and features. Moreover, the study emphasizes the importance of interpretability by analyzing feature importance within each model. This analysis will provide valuable insights that can help cybersecurity professionals understand the factors driving model predictions, making machine learning models more transparent and actionable.

By providing a comprehensive evaluation of different machine learning models, this research aims to offer practical recommendations for organizations seeking to implement machine learning solutions in their cybersecurity infrastructures. The findings of this study will guide decision-makers in selecting the most appropriate models based on their specific needs, whether they prioritize accuracy, speed, or explainability. The remainder of this paper is organized as follows. Section 2 outlines the methodology, including data preparation, feature engineering, and model evaluation techniques. Section 3 presents the experimental results, comparing the performance of different models on the cybersecurity dataset. Finally, Section 4 concludes the paper, summarizing the contributions and proposing directions for future research.

2. RESEARCH METHODOLOGY

The methodology for this study is constructed to evaluate the performance of several machine learning models in detecting and classifying cyberattacks. The approach encompasses dataset preparation, data preprocessing, model development, and evaluation, as well as the application of cross-validation techniques to ensure model robustness.

2.1 Dataset Preparation

The dataset consists of 40,000 records with 25 features, each representing characteristics of network traffic. Data also can be downloaded from [25] Denote the dataset by the feature matrix ($X \in R^{n \times m}$), where (n) is the number of samples, and (m) represents the number of features. The target vector ($y \in \{0,1,2\}^n$) corresponds to the severity levels of the attacks, with labels for high, medium, and low severity. The features include both continuous and categorical variables, such as timestamp, source and destination IP addresses, protocol, packet length, anomaly scores, and attack signatures. In the preprocessing phase, missing data is handled using imputation techniques. For continuous variables, such as packet length, the missing values are replaced by the mean or median of the feature as presented as

$$x_{ij} = \frac{1}{n}\sum_{i=1}^{n} x_{ij}, \quad \text{if} \quad _{ij} \in R \text{ or by the median} \tag{1}$$

where applicable categorical variables are imputed with the mode. Categorical variables, such as protocol type, are encoded using one-hot encoding. Let (C_k) represent a categorical variable with (K) distinct categories. The one-hot encoding transforms (C_k) into a (K)-dimensional vector as presented as

$$C_k \rightarrow C_k' = \{(1,0,\ldots,0),\ldots,(0,\ldots,1)\} \tag{2}$$

Journal of Information System Research (JOSH)
Volume 6, No. 1, Oktober 2024, pp 619−628
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v6i1.6089

2.2 Data Preprocessing

The target labels (y) are encoded using label encoding, mapping the categorical severity levels to numerical values ($\{0,1,2\}$). Continuous features, such as packet length and anomaly scores, are standardized to have zero mean and unit variance as described as

$$x'_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j} \quad (3)$$

where (μ_j) and (σ_j) are the mean and standard deviation of feature (j) respectively. The dataset exhibits class imbalance, where benign traffic outweighs malicious traffic. To mitigate this imbalance, the Synthetic Minority Over-sampling Technique (SMOTE) is employed. SMOTE generates synthetic samples for minority classes by interpolating between existing minority class samples. For a minority class sample (x_i), a synthetic sample (x') is generated as

$$x' = x_i + \lambda(x_j - x_i) \quad (4)$$

where (x_j) is a randomly chosen nearest neighbor of (x_i), and $(\lambda \in [0,1])$. In model development, this study evaluates several machine learning models, including RandomForest, GradientBoosting, XGBoost, LightGBM, CatBoost, and Support Vector Classifier (SVC). The RandomForest model constructs an ensemble of decision trees, where each tree is trained on a random subset of the dataset. The final prediction of the ensemble is given by

$$H(x) = \frac{1}{T}\sum_{t=1}^{T} h(x, \theta_t) \quad (5)$$

where (T) is the total number of trees and (θ_t) represents the parameters for tree (t).

2.3 Machine Learning Models

Boosting models, including GradientBoosting, XGBoost, and LightGBM, iteratively fit weak learners to minimize a loss function. At each stage, a new model $(f_m(x))$ is added to minimize the objective

$$L_m = \sum_{i=1}^{n}\left(y_i - F_{m-1}(x_i) - f_m(x_i)\right)^2 \quad (6)$$

where $(F_m(x) = F_{m-1}(x) + f_m(x))$. In XGBoost, regularization terms are added to the objective function to penalize model complexity

$$Obj = \sum_{i=1}^{n} L(y_i, \hat{y_i}) + \sum_{k=1}^{K} \Omega(f_k) \quad (7)$$

where$(\Omega(f_k))$ controls the complexity of the model. For Support Vector Classifier (SVC), the model finds the optimal hyperplane that maximally separates the classes. This is done by solving the optimization problem

$$\min_{w,b} \quad \frac{1}{2}||w||^2 \quad \text{subject to} \quad y_i(w \cdot x_i + b) \geq 1, \ \forall i \quad (8)$$

This optimization ensures that the hyperplane provides maximum margin separation between the classes. Subsections 2.3.1 – 2.3.7 are explaining the implementation of implemented models into our dataset.

2.3.1 Gradient Boosting Models

In Gradient Boosting, the model operates by sequentially adding weak learners, typically decision trees, to correct the errors of the existing ensemble. The process minimizes a specified loss function, usually mean squared error for regression or log-loss for classification, by adding a new model, denoted $(f_m(x))$, at each stage (m) to predict the residuals, or errors, of the previous prediction $(F_{m-1}(x))$. The objective for each step is to minimize the following function

$$[L_m = \sum_{i=1}^{n}\left(y_i - F_{m-1}(x_i) - f_m(x_i)\right)^2,] \quad (9)$$

where $(F_m(x) = F_{m-1}(x) + f_m(x))$, and (y_i) is the true label for the (i)-th sample. By focusing on the residuals $(\left(y_i - F_{m-1}(x_i)\right))$, the new model $(f_m(x))$ is effectively learning to correct the errors of the ensemble up to that point. This process is repeated until a set number of models are added or a convergence criterion is met. Even if GradientBoosting can be implemented to our case, however, it has potential to face generalization problem on our cybersecurity data, likely due to challenges in learning intricate patterns amid the noise. In addition, while it avoids extreme overfitting, it has potential to fails to sufficiently capture the minority class patterns.

2.3.2 XGBoost

XGBoost, or Extreme Gradient Boosting, builds on the principles of Gradient Boosting by incorporating additional regularization terms to control the complexity of each added model. This is particularly relevant for datasets with a high risk of overfitting due to noisy or high-dimensional features, as is often the case in cybersecurity datasets.

Journal of Information System Research (JOSH)
Volume 6, No. 1, Oktober 2024, pp 619−628
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v6i1.6089

The objective function for XGBoost incorporates a regularization term ($\Omega(f_k)$), which penalizes the complexity of the model such as

$$[\text{Obj} = \sum_{i=1}^{n} L(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)\,,] \quad (10)$$

where ($L(y_i, \hat{y}_i)$) is the loss function measuring the discrepancy between the true labels (y_i) and the predicted values ($\hat{y}_i$), and ($\Omega(f_k)$) controls the complexity of each weak learner (f_k) in the model ensemble. Regularization terms, such as L1 (Lasso) and L2 (Ridge) penalties, help prevent overfitting by discouraging the model from growing overly complex and fitting to noise rather than meaningful patterns in the data. In cybersecurity datasets, where classes are often imbalanced, regularization becomes particularly beneficial, as it keeps the model from over-adjusting to dominant (benign) patterns and ignoring minority (malicious) cases. XGBoost also uses optimizations like parallel processing, tree pruning, and cache-awareness, which enhance performance and make it well-suited for large datasets. Even if the XGBoost's can be implemented to our case, but there is a limitation such as inability to generalize effectively that may be partly due to its susceptibility to noise within minority classes, despite regularization.

2.3.3 Light GBM

LightGBM (Light Gradient Boosting Machine) is another variant of Gradient Boosting that offers efficiency improvements. It uses a unique leaf-wise growth strategy that expands the tree nodes with the maximum reduction in loss rather than growing all nodes level-by-level. This strategy allows LightGBM to achieve higher accuracy and faster training times, especially on large datasets. Given that LightGBM also incorporates regularization, it is less likely to overfit to the training data than traditional Gradient Boosting. In order to implemen LightGBM to handles categorical features natively, the avoiding the need for one-hot encoding strategy is conducted, which helps streamline the preprocessing pipeline.

2.3.4 Support Vector Classifier (SVC)

The Support Vector Classifier (SVC) operates on a fundamentally different principle from boosting models. Instead of sequentially improving on errors, SVC seeks to find an optimal hyperplane that separates data points of different classes with the maximum margin. This separation maximizes the distance (or "margin") between the classes, providing a robust decision boundary. SVC accomplishes this by solving the following optimization problem

$$[\min_{w,b} \quad \frac{1}{2}|w|^2 \quad \text{subject to} \quad y_i(w \cdot x_i + b) \geq 1,\ \forall i,] \quad (11)$$

where (w) represents the weights of the model, (b) is the intercept, and (y_i) denotes the class label for each data point (x_i). The objective of this optimization is to minimize the norm of (w), effectively finding the smallest weights necessary to separate the classes, while ensuring all data points are on the correct side of the hyperplane.

In high-dimensional, complex datasets, such as cybersecurity data, SVC can struggle if there is a high degree of overlap or noise in the features, as is often the case. In these instances, SVC's performance can be improved by kernelizing the algorithm, using methods like the radial basis function (RBF) kernel, which maps the data into a higher-dimensional space where it becomes more separable. In our cybersecurity dataset, these models exhibit varied performance based on their handling of high-dimensional, imbalanced data. SVC, which finds the maximum-margin hyperplane, can be struggled with the noisy and overlapping feature space. Therefore, the Hyperparameter tuning strategy can be implemented, especially for boosting models, it could also enhance model accuracy by optimizing learning rate, tree depth, and regularization terms. Addressing class imbalance through methods like SMOTE or cost-sensitive learning will be essential in improving these models' ability to detect minority classes accurately in cybersecurity contexts.

2.3.5 CatBoost

CatBoost is a gradient boosting algorithm that efficiently manages categorical data and is designed to handle complex datasets, like those found in cybersecurity, with minimal preprocessing. CatBoost minimizes a loss function ($L\big(y, f(x)\big)$) iteratively, where each boosting iteration seeks to reduce the residual errors from the previous model. Initially, the model sets a baseline by finding the constant that minimizes the loss over all samples. Then, in each iteration, the algorithm calculates residuals as the negative gradient of the loss with respect to the model's output, fits a decision tree to these residuals, and updates the model by adding the new tree's predictions scaled by a learning rate, \(\eta \). For classification tasks, CatBoost often uses Log Loss as the objective, defined as

$$[L\big(y, f(x)\big) = -\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)]] \quad (12)$$

where (y_i) represents the true label, and ($\hat{y}_i$) is the predicted probability. This iterative approach allows CatBoost to capture complex relationships in high-dimensional data, making it suitable for identifying subtle patterns in cybersecurity datasets with categorical and numerical features.

Journal of Information System Research (JOSH)
Volume 6, No. 1, Oktober 2024, pp 619−628
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v6i1.6089

2.3.6 Random Forest

Random Forest is an ensemble learning technique that constructs multiple decision trees using random subsets of data and features, then combines their predictions. Each tree in the forest is built on a bootstrap sample such as a random sample of the data with replacement. At each split within a tree, only a subset of features is randomly selected to determine the best split, which helps reduce correlation among trees. Each tree in the Random Forest is trained using a criterion like Gini Index, which measures node impurity by calculating

$$[\text{Gini} = 1 - \sum_{c=0}^{1} p_c^2] \quad (13)$$

where (p_c) is the proportion of samples of class (c) at a node. For predictions, Random Forest uses a majority vote over the predictions from all trees in the ensemble. This approach is well-suited to high-dimensional cybersecurity data as it reduces the likelihood of overfitting, balancing between capturing patterns and generalizing to unseen data.

2.3.7 Logistic Regression

Logistic Regression is a linear classification model often used as a baseline in binary classification tasks. It calculates the probability of a class based on a linear combination of input features. For a given instance, the model computes

$$[P(y = 1|x) = \frac{1}{1+e^{-(\beta_0+\beta^T x)}}] \quad (14)$$

where (β) is the vector of coefficients for each feature, and (β_0) is the intercept. To train the model, Logistic Regression minimizes Binary Cross-Entropy Loss such as

$$[L(\beta) = -\sum_{i=1}^{N}[y_i \log(P(y = 1|x_i)) + (1 - y_i)\log(1 - P(y = 1|x_i))]] \quad (15)$$

this loss function is minimized using gradient descent, where the gradient of the loss with respect to (β) is $[\nabla_\beta L(\beta) = \sum_{i=1}^{N}(P(y = 1|x_i) - y_i)x_i]$ though Logistic Regression may be limited when dealing with highly non-linear, high-dimensional cybersecurity data, it provides a straightforward baseline that can help assess the linear separability of the dataset before implementing more complex models.

2.4. Evaluation Metrics

To evaluate the models, several metrics are used, including accuracy, precision, recall, and F1-score. Let the confusion matrix (M) represent the model's predictions, where (M_{ij}) represents the number of instances with true label (i) and predicted label (j). Accuracy is defined as

$$\text{Accuracy} = \frac{\sum_i M_{ii}}{\sum_i \sum_j M_{ij}} \quad (16)$$

while precision and recall are calculated as

$$\text{Precision} = \frac{M_{ii}}{\sum_j M_{ij}} \quad (17)$$

$$\text{Recall} = \frac{M_{ii}}{\sum_i M_{ij}} \quad (18)$$

F1-score is the harmonic mean of precision and recall

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (19)$$

Cross-validation is applied to ensure the generalization of the models. Stratified (k)-fold cross-validation is used, preserving the class distribution across folds. Given (D), the dataset partitioned into (k) folds, the model is trained on $(k-1)$ folds and validated on the remaining fold. The process is repeated (k) times, and the average performance across folds is computed as

$$\text{Perf} = \frac{1}{k}\sum_{j=1}^{k} P(D_j) \quad (20)$$

where $(P(D_j))$ is the performance on fold (j). For interpretability, feature importance is extracted for tree-based models like RandomForest and XGBoost. Let $(I(f_i))$ represent the importance of feature (f_i), computed as

$$I(f_i) = \sum_{t=1}^{T} \sum_{n \in N_t} \Delta G(f_i, n) \quad (21)$$

where $(\Delta G(f_i, n))$ is the reduction in Gini impurity from splitting on feature (f_i) at node (n) in tree (t).

Journal of Information System Research (JOSH)
Volume 6, No. 1, Oktober 2024, pp 619−628
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v6i1.6089

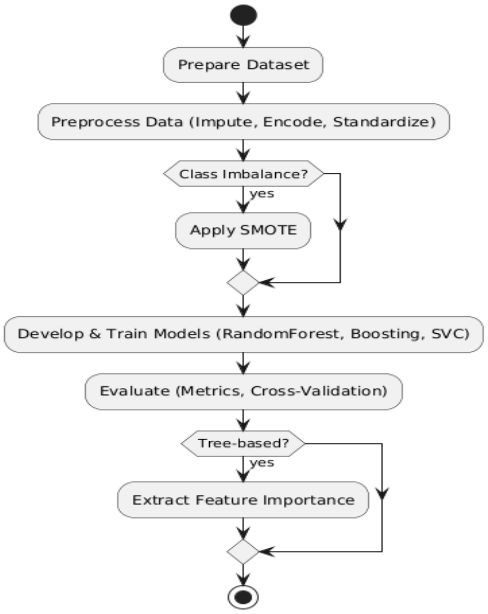

Figure 1. Experiment Process

3. RESULT AND DISCUSSION

3.1 Result

The evaluation of several machine learning models, including RandomForest, GradientBoosting, XGBoost, LightGBM, CatBoost, Support Vector Classifier (SVC), Logistic Regression, and a Voting Classifier, provided valuable insights into their performance on the cybersecurity dataset as presented in the table 1. Each model was trained and tested, and the results showed substantial variation in their ability to generalize from the training data to the test data. The RandomForest model achieved a high training accuracy of 99.84%, indicating that the model was able to learn from the training data very well. However, this high performance did not carry over to the test set, where the accuracy dropped significantly to 33.77%. This large gap between the training and test accuracies points to overfitting, where the model learned patterns in the training data, including noise, but struggled to generalize to new, unseen data. This behavior is characteristic of RandomForest when it is trained on complex, high-dimensional datasets without sufficient regularization.

In contrast, the GradientBoosting model displayed a training accuracy of 40.26% and a test accuracy of 33.50%. While the gap between the training and test accuracies is smaller compared to RandomForest, GradientBoosting's overall performance remained low. The model struggled to find meaningful patterns in the dataset, as evidenced by its suboptimal training accuracy. The test accuracy being almost identical to the training accuracy suggests that the model did not overfit but also did not capture enough information to distinguish between the different classes effectively. XGBoost, another boosting algorithm, demonstrated a better fit to the training data, achieving a training accuracy of 62.11%. However, its test accuracy remained low at 33.49%, similar to GradientBoosting. Despite its higher training accuracy, the model's inability to generalize suggests that it may have started overfitting to the training set but was not able to apply the learned patterns to the test set effectively. This result indicates that while XGBoost can capture more patterns than GradientBoosting, it still faces challenges with generalization on this dataset.

Table 1. Machine Learning Performance

Model	Train Accuracy	Test Accuracy
RandomForest	0.9984	0.3377
GradientBoosting	0.4026	0.335
XGBoost	0.6211	0.3349
LightGBM	0.5341	0.336
CatBoost	0.5689	0.3338
SVC	0.4355	0.3366
LogisticRegression	0.3415	0.3301
VotingClassifier	0.9742	0.3356

LightGBM, known for its efficiency, produced a training accuracy of 53.41%, which is slightly lower than XGBoost but higher than GradientBoosting. Its test accuracy, 33.60%, was comparable to the other models.

Journal of Information System Research (JOSH)
Volume 6, No. 1, Oktober 2024, pp 619−628
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v6i1.6089

LightGBM's leaf-wise tree growth likely helped it capture some patterns in the dataset, but the relatively low test accuracy indicates that, like the other models, it struggled to generalize effectively to the test data. The CatBoost model achieved a training accuracy of 56.89% and a test accuracy of 33.38%. This model is designed to handle categorical features without the need for extensive preprocessing, which may have contributed to its relatively high training accuracy. However, its test accuracy suggests that it also faced similar challenges to the other boosting models when it came to generalization.

The SVC model produced a training accuracy of 43.55% and a test accuracy of 33.66%. SVC, which attempts to find an optimal hyperplane to separate the classes, struggled with the complexity and high dimensionality of the dataset. The low accuracies suggest that SVC may not have been able to find a clear boundary between the classes, particularly in the presence of noisy or overlapping features. Logistic Regression, often used as a baseline model for classification tasks, performed the worst among the models. It achieved a training accuracy of 34.15% and a test accuracy of 33.01%. This result is unsurprising, given that Logistic Regression assumes a linear relationship between the features and the target, which is unlikely to hold in a complex, high-dimensional dataset like this one. Finally, the Voting Classifier, which combined the predictions of RandomForest, XGBoost, LightGBM, and CatBoost, achieved a high training accuracy of 97.42%, but its test accuracy remained low at 33.56%. This ensemble approach was expected to leverage the strengths of the individual models, but the low test accuracy suggests that the ensemble did not effectively mitigate the generalization issues observed in the individual models.

3.2 Discussion

The consistently low-test accuracies across all models highlight several key challenges inherent in the dataset and the modeling process. One of the most significant factors affecting performance is the imbalance in the dataset. The benign traffic, which constitutes the majority class, dominates the data, making it difficult for the models to detect and classify cyberattacks, which represent the minority class. This imbalance likely caused the models to focus more on the majority class, leading to poor performance in identifying malicious traffic. The tendency to favor the majority class is particularly evident in the test results, where none of the models was able to achieve a test accuracy higher than 33.77%, despite some models showing high performance on the training data. Overfitting emerged as a major issue for many of the models, particularly RandomForest and the ensemble Voting Classifier. RandomForest's extremely high training accuracy of 99.84% suggests that the model memorized the training data, learning not only the underlying patterns but also the noise. This behavior, typical of models with high capacity and little regularization, is problematic because it results in poor generalization to new data, as evidenced by the significant drop in test accuracy. The Voting Classifier, despite combining multiple models, exhibited a similar overfitting pattern, with a training accuracy of 97.42% but a test accuracy of only 33.56%. These results suggest that the ensemble model was not able to overcome the limitations of its individual components, and the issue of overfitting persisted.

The boosting models such as GradientBoosting, XGBoost, LightGBM, and CatBoost showed more moderate levels of overfitting, with smaller gaps between their training and test accuracy. However, these models also struggled to generalize, as evidenced by their low-test accuracies, which were all around 33%. This suggests that while boosting algorithms are typically better at reducing bias and improving generalization, the complexity and noise in the dataset limited their effectiveness in this case. It is also worth noting that boosting algorithms are sensitive to hyperparameters, and further tuning of the learning rate, number of estimators, and regularization parameters could potentially improve their performance. Another important factor that likely contributed to the models' poor performance is the high dimensionality of the dataset. High-dimensional data often contains irrelevant or noisy features, which can confuse machine learning models and prevent them from learning meaningful patterns. In this case, the dataset included 25 features, some of which may not have been informative for the task of detecting cyberattacks. The presence of noisy features could have contributed to the models' difficulty in generalizing from the training data to the test data. Feature selection techniques, such as Recursive Feature Elimination (RFE), could be used in future work to reduce the feature space and focus the models on the most relevant features. Dimensionality reduction methods like Principal Component Analysis (PCA) could also be employed to project the data into a lower-dimensional space where the relationships between features are more apparent.

Handling class imbalance is another crucial area for future improvement. The imbalanced nature of the dataset likely skewed the models' predictions toward the majority class, resulting in poor performance on the minority class (cyberattacks). Techniques such as oversampling the minority class, undersampling the majority class, or generating synthetic data through methods like the Synthetic Minority Over-sampling Technique (SMOTE) could help mitigate this issue. Cost-sensitive learning, where the models assign a higher cost to misclassifying the minority class, could also improve the detection of cyberattacks. Adjusting the decision threshold to prioritize the detection of minority class instances could further enhance performance. Finally, hyperparameter tuning is an essential step for improving the models' performance. Many of the models, particularly the boosting algorithms, are highly sensitive to hyperparameters such as the learning rate, number of estimators, tree depth, and regularization terms. A more exhaustive search of the hyperparameter space, using techniques such as grid search or randomized search, could help optimize the models and potentially lead to better

Journal of Information System Research (JOSH)
Volume 6, No. 1, Oktober 2024, pp 619−628
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v6i1.6089

generalization. Additionally, regularization techniques, such as L1 or L2 regularization, could help prevent overfitting by penalizing overly complex models. In conclusion, while the models demonstrated varying degrees of success in fitting the training data, they all struggled to generalize to the test data, as evidenced by their low-test accuracies. The primary factors contributing to this poor generalization include the class imbalance, high dimensionality, and noise in the dataset, as well as potential overfitting in some of the models. Future work should focus on addressing these issues through more sophisticated data preprocessing, feature selection, class balancing, and hyperparameter tuning to improve the models' ability to detect and classify cyberattacks in real-world settings.

3.3 Internal Validity

One significant threat to internal validity is the issue of class imbalance. The dataset used in this study was heavily imbalanced, with benign traffic vastly outnumbering malicious cyberattack instances. This imbalance likely skewed the models' learning process, making it more difficult for them to correctly identify the minority class (cyberattacks). Although the class imbalance was recognized, the steps taken to mitigate its impact, such as using default machine learning models, were limited. More sophisticated techniques, such as oversampling, undersampling, or cost-sensitive learning, were not fully implemented, which could have improved the models' ability to handle this imbalance. Another potential threat is overfitting, particularly in models like RandomForest and the Voting Classifier, which exhibited high training accuracies but poor generalization to the test data. The overfitting observed in these models suggests that the models may have memorized the training data, including noise, rather than learning generalizable patterns. This threatens the internal validity because it indicates that the models may not be accurately capturing the true relationships in the data. Overfitting could have been reduced with more robust regularization techniques, hyperparameter tuning, or by increasing the diversity of training data

A related internal threat is the lack of comprehensive hyperparameter tuning. While some hyperparameter optimization was performed, the tuning was relatively limited in scope. Models like XGBoost, LightGBM, and GradientBoosting are highly sensitive to hyperparameters such as learning rate, the number of estimators, and tree depth. Without an exhaustive search through the hyperparameter space, there is a risk that the models were not operating in their most optimal configuration, potentially limiting their performance. The dimensionality of the dataset also presents a threat to internal validity. The dataset contained 25 features, and it is possible that some of these features were irrelevant or noisy. High-dimensional datasets are prone to the "curse of dimensionality," where irrelevant features can obscure the meaningful patterns that models need to learn. While the models were able to handle these dimensions, the lack of feature selection or dimensionality reduction methods may have prevented the models from focusing on the most important features. This could have contributed to the models' poor performance, and future studies should consider using feature engineering or selection techniques to improve internal validity.

3.4 External Validity

One major threat to external validity is the generalizability of the dataset. The dataset used in this study, while representative of a real-world cybersecurity environment, may not fully reflect the variety and complexity of cyberattacks encountered in different domains or organizations. Cybersecurity attacks are continuously evolving, with new attack vectors and strategies being developed by malicious actors. Thus, a model that performs poorly on this specific dataset might not necessarily behave the same way on a different dataset with different attack patterns. Moreover, the dataset's traffic patterns may be specific to a particular network setup or environment, limiting the models' applicability to other network infrastructures or attack scenarios. Another threat to external validity is the choice of machine learning models. While a range of models, including RandomForest, GradientBoosting, XGBoost, and LightGBM, were tested, more advanced models such as deep learning architectures were not explored. Given the complex and high-dimensional nature of cybersecurity data, deep learning models like convolutional neural networks (CNNs) or recurrent neural networks (RNNs) may be better suited to capture the intricacies of cyberattacks. The absence of such models in this study means that the results may not generalize well to experiments using more sophisticated algorithms.

3.5 Practical Implications

The findings of this study suggest several practical implications for cybersecurity practitioners and data scientists working with machine learning models in cybersecurity. First, the overfitting issue observed, especially in models like RandomForest and the Voting Classifier, underscores the need for rigorous regularization and hyperparameter tuning when working with high-dimensional, noisy datasets. Organizations implementing machine learning-based cybersecurity solutions should invest in extensive preprocessing techniques, including feature selection and dimensionality reduction, to optimize model performance and minimize overfitting. Additionally, the issue of class imbalance is common in cybersecurity data where benign traffic outweighs malicious instances, it highlights the need for advanced class-balancing techniques. By incorporating approaches such as SMOTE, undersampling, or cost-sensitive learning, practitioners can improve model sensitivity to minority classes, which is critical for accurately detecting cyberattacks in real-time systems. Furthermore, employing models specifically designed for high-dimensional, imbalanced data, or exploring ensemble methods with weighted voting or customized decision

Journal of Information System Research (JOSH)
Volume 6, No. 1, Oktober 2024, pp 619−628
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v6i1.6089

thresholds, may provide better detection rates in complex cybersecurity scenarios. Finally, given that cybersecurity threats evolve rapidly, this study points to the value of continuously updating models and datasets to incorporate the latest attack vectors. Organizations should consider implementing adaptive machine learning models or periodic retraining schedules to maintain model relevancy and performance, enhancing the overall robustness of cybersecurity defenses.

4. CONCLUSION

This study provides an in-depth evaluation of multiple machine learning models in the detection and classification of cybersecurity attacks, demonstrating critical insights into the performance of RandomForest, GradientBoosting, XGBoost, LightGBM, CatBoost, SVC, Logistic Regression, and Voting Classifiers on a high-dimensional, real-world dataset. Although RandomForest and the Voting Classifier exhibited high accuracy on the training set, significant overfitting was observed, indicating that these models may have memorized patterns specific to the training data without capturing generalizable features relevant to the test data. Meanwhile, GradientBoosting and XGBoost showed moderate overfitting, yet still struggled to effectively generalize, emphasizing the need for targeted preprocessing, including feature selection and dimensionality reduction, to improve model accuracy and reliability. The study's findings also reveal the challenges posed by class imbalance, with the models gravitating toward majority (benign) traffic and underperforming on the minority (malicious) class. This limitation underscores the importance of exploring advanced class-balancing methods to improve the detection rates for cybersecurity applications. Moving forward, the study suggests that future research should incorporate more sophisticated preprocessing techniques, hyperparameter tuning, and model architectures potentially extending to deep learning approaches like CNNs or RNNs to better handle the complexities of high-dimensional and imbalanced cybersecurity data. This would enhance the models' applicability in real-world cybersecurity settings, contributing to more robust and adaptable threat detection systems capable of meeting the dynamic demands of modern cyber defenses.

REFERENCES

[1] C.-L. Chen, Y.-C. Lin, W.-H. Chen, C.-F. Chao, and H. Pandia, “Role of government to enhance digital transformation in small service business,” Sustainability, vol. 13, no. 3, p. 1028, 2021.
[2] S. Kraus, P. Jones, N. Kailer, A. Weinmann, N. Chaparro-Banegas, and N. Roig-Tierno, “Digital transformation: An overview of the current state of the art of research,” Sage Open, vol. 11, no. 3, p. 21582440211047576, 2021.
[3] T. Saarikko, U. H. Westergren, and T. Blomquist, “Digital transformation: Five recommendations for the digitally conscious firm,” Bus. Horiz., vol. 63, no. 6, pp. 825–839, 2020.
[4] D. J. Edwards, “Malware Defenses,” in Critical Security Controls for Effective Cyber Defense: A Comprehensive Guide to CIS 18 Controls, Springer, 2024, pp. 277–308.
[5] M. Huszár, “Current state of IT security awareness--challenges, risks and effects globally.”
[6] M. Anisetti, C. Ardagna, M. Cremonini, E. Damiani, J. Sessa, and L. Costa, “Security threat landscape,” White Pap. Secur. Threat., 2020.
[7] S. S. M. Dandyala and S. Banik, “Traditional Methods of Threat Detection,” Int. J. Adv. Eng. Technol. Innov., vol. 1, no. 2, pp. 161–177, 2021.
[8] K. Hamid, M. W. Iqbal, M. Aqeel, X. Liu, and M. Arif, “Analysis of Techniques for Detection and Removal of Zero-Day Attacks (ZDA),” in International Conference on Ubiquitous Security, 2022, pp. 248–262.
[9] I. H. Sarker, “CyberLearning: Effectiveness analysis of machine learning security modeling to detect cyber-anomalies and multi-attacks,” Internet of Things, vol. 14, p. 100393, 2021.
[10] L. Cui, Y. Qu, L. Gao, G. Xie, and S. Yu, “Detecting false data attacks using machine learning techniques in smart grid: A survey,” J. Netw. Comput. Appl., vol. 170, p. 102808, 2020.
[11] C. Xenofontos, I. Zografopoulos, C. Konstantinou, A. Jolfaei, M. K. Khan, and K.-K. R. Choo, “Consumer, commercial, and industrial iot (in) security: Attack taxonomy and case studies,” IEEE Internet Things J., vol. 9, no. 1, pp. 199–221, 2021.
[12] S. A. Toledano, Critical Infrastructure Security: Cybersecurity lessons learned from real-world breaches. Packt Publishing Ltd, 2024.
[13] R. S. Gonzalez, R. A. da Silveira Rossi, and L. G. M. Vieira, “Economic and financial consequences of process accidents in Brazil: Multiple case studies,” Eng. Fail. Anal., vol. 132, p. 105934, 2022.
[14] T. Sobb, B. Turnbull, and N. Moustafa, “Supply chain 4.0: A survey of cyber security challenges, solutions and future directions,” Electronics, vol. 9, no. 11, p. 1864, 2020.
[15] T. Anitha, S. Aanjankumar, S. Poonkuntran, and A. Nayyar, “A novel methodology for malicious traffic detection in smart devices using BI-LSTM--CNN-dependent deep learning methodology,” Neural Comput. Appl., vol. 35, no. 27, pp. 20319–20338, 2023.
[16] A. Muttepawar, “Detecting Distributed Denial of Service attack using ensemble learning,” Dublin, National College of Ireland, 2021.
[17] L. Liu, P. Wang, J. Lin, and L. Liu, “Intrusion detection of imbalanced network traffic based on machine learning and deep learning,” IEEE access, vol. 9, pp. 7550–7563, 2020.
[18] L. Sun, Y. Zhou, Y. Wang, C. Zhu, and W. Zhang, “The effective methods for intrusion detection with limited network attack data: Multi-task learning and oversampling,” IEEE access, vol. 8, pp. 185384–185398, 2020.
[19] S. Han, C. Lin, C. Shen, Q. Wang, and X. Guan, “Interpreting adversarial examples in deep learning: A review,” ACM

Journal of Information System Research (JOSH)
Volume 6, No. 1, Oktober 2024, pp 619−628
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v6i1.6089

Comput. Surv., vol. 55, no. 14s, pp. 1–38, 2023.
[20] P. Linardatos, V. Papastefanopoulos, and S. Kotsiantis, “Explainable ai: A review of machine learning interpretability methods,” Entropy, vol. 23, no. 1, p. 18, 2020.
[21] G. Agrawal, A. Kaur, and S. Myneni, “A review of generative models in generating synthetic attack data for cybersecurity,” Electronics, vol. 13, no. 2, p. 322, 2024.
[22] O. Serradilla, E. Zugasti, J. Rodriguez, and U. Zurutuza, “Deep learning models for predictive maintenance: a survey, comparison, challenges and prospects,” Appl. Intell., vol. 52, no. 10, pp. 10934–10964, 2022.
[23] L. E. Lwakatare, A. Raj, I. Crnkovic, J. Bosch, and H. H. Olsson, “Large-scale machine learning systems in real-world industrial settings: A review of challenges and solutions,” Inf. Softw. Technol., vol. 127, p. 106368, 2020.
[24] R. Ahmad and I. Alsmadi, “Machine learning approaches to IoT security: A systematic literature review,” Internet of Things, vol. 14, p. 100365, 2021.
[25] T. Incribo, "Cyber Security Attacks," Kaggle, 2023. [Online]. Available: https://www.kaggle.com/datasets/teamincribo/cyber-security-attacks. [Accessed: 1-Oct-2024].