MOBILE
AND FORENSICS

ISSN 2656-6257
Vol. 8, No. 1, March 2026, pp. 1-12

Performance Analysis of Random Forest Algorithm with SMOTE for Multi-Class Attack Detection

Ratna Komalasari[a,1,*], Mukhlis Prasetyo Aji[a,2], Agung Purwo Wicaksono[a,3], Maulida Ayu Fitriani[a,4]

[a]Teknik Informatika, Universitas Muhammadiyah Purwokerto, Banyumas, Indonesia
[1]2003040086@ump.ac.id, [2]Pasetyo-aji@ump.ac.id, [3]maulidaayuf@ump.ac.id, [4]wicaksono@ump.ac.id
* corresponding author

ARTICLE INFO

Article history
Received : October 1, 2025
Revised : October 26, 2025
Accepted : November 17, 2025

Keywords
cyberattack detection,
multi-class classification,
random forest,
SMOTE,
stratified random sampling

ABSTRACT

The increasing sophistication of cyberattacks necessitates the development of detection systems capable of accurately identifying various threat types. Data imbalance within attack logs presents a substantial challenge that can undermine the effectiveness of detection models. This study introduces a multi-class cyberattack detection model employing the Random Forest algorithm, optimized through the Synthetic Minority Over-sampling Technique (SMOTE) to address data imbalance. The innovative aspect of this research lies in integrating Random Forests and SMOTE to improve multi-class classification accuracy on local attack log datasets. This approach remains sparsely explored in academic research. The dataset consists of 3000 cyberattack logs from the Information Systems Bureau of Muhammadiyah University Purwokerto, spanning 10 cyberattack categories. The research process involved data collection, pre- processing, division, model training, and evaluation. Results indicate that the model achieved an average F1-macro score of 76% and a weighted average of 93%, with the " Threat Level Medium " feature identified as the most influential predictor. These findings suggest that the combination of Random Forest and SMOTE effectively enhances multi-class detection performance and presents promising prospects for log-based cybersecurity systems in educational and industrial environments.

1. Introduction

Cybersecurity attacks are escalating and have emerged as a significant concern across diverse sectors, including government agencies, financial institutions, and critical infrastructure. The increasing intricacy of cyberattacks has rendered it more challenging for traditional systems to identify dynamic and adaptive threats effectively[1]. In Indonesia, cyberattacks have exhibited a consistent annual increase. According to a report by the National Cyber and Crypto Agency (BSSN), in 2024, the number of suspicious cyberattack anomalies exceeded 330 million, reflecting a 40% increase from the previous year [2].

Anomalies in cybersecurity pertain to patterns or activities that diverge from standard operational parameters within a system or network. Such anomalies may function as preliminary indicators of prospective cyber threats. The various categories of cyberattacks encompass backdoor intrusions, information disclosure, operating system command injections, WebShell uploads, website scanning, manipulation of URL access, circumvention of request method filters, inspection of web access in

https://doi.org/10.12928/mf.v8i1.14584

cleartext requests, path traversal, and constraints on file downloads. [3]. These assaults jeopardize the integrity and security of information systems. Detecting these ten attack categories concurrently poses a complex multi-class classification challenge, especially when data distributions are substantially imbalanced. This research utilizes a dataset comprising 3,000 cyberattack logs from the Information Systems Bureau of Muhammadiyah University Purwokerto. The dataset was chosen because it encompasses diverse modern attack types and accurately represents the real-world data imbalance typically observed in network security systems. [4].

This severe imbalance constitutes the primary challenge that distinguishes this study from previous research, which generally employs benchmark datasets with more balanced distributions. This investigation uses a machine learning methodology, specifically the Random Forest algorithm, to address the complexity of multi-class cyberattack detection. The choice of this algorithm is based on its ability to perform classification by analyzing feature importance among the most relevant attributes [5]. Furthermore, stratified random sampling is employed to ensure representativeness within each attack category. Conversely, SMOTE (Synthetic Minority Over-sampling Technique) is used to balance class distributions in the training dataset. Several prior studies have applied Random Forest and SMOTE in the context of intrusion detection, albeit with varying focuses and limitations. For instance, a review by Ikhwanul Uzlah et al. (2024) indicated that Random Forest achieved an accuracy of 66% but was primarily restricted to binary classification (attack versus normal) [6]. The research conducted by Wu et al. (2022) integrated an advanced Random Forest with the SMOTE methodology and employed the K-means clustering algorithm, achieving classification accuracies of 99.72% on the training dataset and 78.47% on the testing dataset of the NSL-KDD dataset [7]. Although promising, the study experienced a substantial decline in performance when addressing multi-class scenarios with extreme data imbalance, underscoring the need for further optimization. Meanwhile, Talukder et al. (2025) used K-Means-SMOTE in conjunction with Decision Tree and Random Forest models, achieving 99.94% accuracy and 99.94% F1-score on the WSN-DS dataset [8]. However, the focus of this research was on wireless sensor networks and IoT, with traffic patterns different from those in web application logs, so this approach might not be optimal for detecting attacks based on local institutional logs.

Based on this review, this study highlights three main research gaps. First, there is a lack of research specifically addressing extreme class imbalance (ratio > 1:200) in multi-class attack detection. Second, there is limited exploration of Random Forest-SMOTE on institutional local log datasets with attack characteristics distinct from those in public benchmark datasets. Third, a comprehensive analysis of how specific features contribute to detecting various types of web attacks is not yet available. The novelty of this research lies in three key aspects. The first is the integration of SMOTE and Random Forest techniques optimized for multi-class classification on datasets with extreme imbalance (ratio 1:238). The second is the application of feature importance analysis to identify critical attributes that differentiate ten specific categories of web attacks. The third is the validation of this approach using local institutional data that reflect the actual conditions of educational infrastructure in Indonesia.

This research aims to develop a precise multi-class cyberattack detection model by optimizing the combination of the most pertinent numerical and categorical features. The Synthetic Minority Over-sampling Technique (SMOTE) is used to balance the training dataset, enabling the model to identify minority attack patterns without compromising the accuracy of the majority class. The study's contributions encompass empirical validation of the effectiveness of the Random Forest-SMOTE approach on limited institutional datasets characterized by significant imbalances, identification of critical features for log-based web attack detection via feature importance analysis, and recommendations for a layered detection architecture suitable for integration into institutional Security Information and Event Management (SIEM) systems. It is anticipated that this research will lay the foundation for a more comprehensive and adaptable cyberattack detection system in educational and organizational environments facing similar attack vectors and resource limitations.

2. Method

This research was conducted utilizing the Random Forest machine learning algorithm. The stages of the study are delineated as follows:

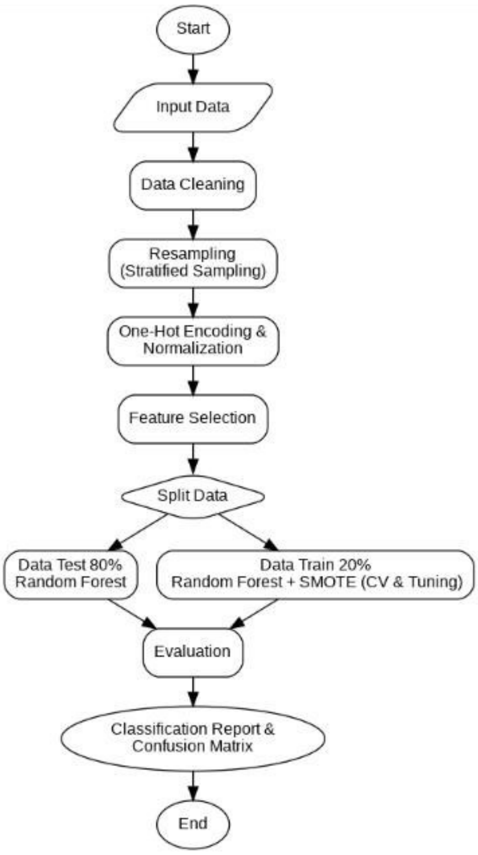

Start
Input Data
Data Cleaning
Resampling (Stratified Sampling)
One-Hot Encoding & Normalization
Feature Selection
Split Data
Data Test 80% Random Forest
Data Train 20% Random Forest + SMOTE (CV & Tuning)
Evaluation
Classification Report & Confusion Matrix
End

Fig. 1. Research Method Flowchart

2.1. Data Collection

This research employed a quantitative methodology with a documentation study approach. Data comprising cyberattack logs in .csv format were sourced from the Information Systems Bureau of Muhammadiyah University Purwokerto. A total of 3000 entries were selected through the Stratified Random Sampling technique to ensure proportional representation of each attack type. [9]. Each attack class was restricted to a maximum of 848 samples, determined by the class with the highest number, to ensure a balanced distribution [10]. While 3,000 samples may be considered relatively modest for generalization, this dataset accurately reflects the authentic circumstances of an educational institution, characterized by specific attack features. The dataset is adequate for preliminary validation of the attack-detection system in a school environment; however, additional data are required to improve the model's generalization.

2.2. Data Exploration

Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract. Abbreviations such as IEEE, SI, MKS, CGS, sc, dc, and rms do not have to be defined. Do not use abbreviations in the title or heads unless they are unavoidable.

2.3. Pre-processing

Data preprocessing is a fundamental stage in machine learning that involves cleaning, transforming, and preparing raw data for subsequent analysis. [12]. This procedure encompasses data cleansing, normalization of numerical attributes, and encoding of categorical attributes. During data cleansing, columns that could induce data leakage, are irrelevant, contain missing values, are duplicates, or do not provide predictive information are eliminated. Numerical attributes are normalized utilizing MinMaxScaler, whereas categorical attributes are encoded via one-hot encoding. Feature selection is conducted based on correlation with the target variable to improve performance and mitigate overfitting. Subsequently, the dataset is partitioned into training (80%) and testing (20%) subsets employing a stratified train-test split. SMOTE (k_neighbors=1) is applied exclusively to the training data to balance the classes and prevent data leakage [13].

2.4. Model Training

The primary model is a Random Forest Classifier, with hyperparameter optimization performed through GridSearchCV and 5-fold cross-validation. The Random Forest algorithm was selected for its reliable performance, high interpretability, and computational efficiency on medium-sized datasets. [14] Compared to boosting-based ensemble methods such as XGBoost and LightGBM, Random Forest exhibits greater robustness against overfitting and features a more straightforward training procedure, while maintaining a comparable level of accuracy [15]. Table 1 presents the parameter ranges tested.

Table 1. GidSearchCV Hyperparameter Configuration

Parameter	Tested Values
n_estimators	100, 200, 300
max_depth	10, 15, 20, None
max_features	sqrt, log2

The primary optimization metric employed is F1-macro [16]. Random Forest trains an ensemble of decision trees on SMOTE-processed data, where each tree learns the feature-target relationship using randomly selected data and feature subsets, resulting in robust predictions that generalize well to test data [17].

2.5. Prediction and Evaluation

During the prediction phase, the trained model categorizes test data into the target class. These predictions are utilized to assess the model's capacity to generalize to novel (unseen) data, based on the patterns acquired during training [7]. The prediction results are subsequently assessed utilizing classification metrics such as accuracy, precision, recall, and F1-score to evaluate the model's capability to categorize each class accurately [18]. A confusion matrix helps evaluate how well the model distinguishes each attack category[19]. 5-fold cross-validation is employed to ensure the stability and reliability of the model's performance consistent [20]. The assessment is performed on test data that the model has not previously encountered during training to evaluate its generalization capability [21]. The formula for the evaluation metric as shown in Fig. 2:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \quad (1)$$

$$Precision = \frac{TP}{FP + TP} \times 100\% \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \times 100\% \quad (3)$$

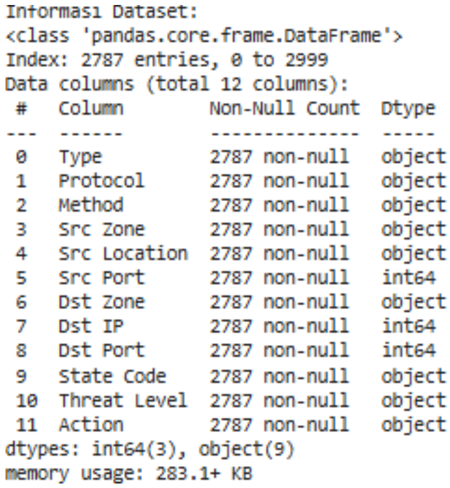
$$F1score = \frac{2 \times (precision \times recall)}{precision + recall} \times 100\% \quad (4)$$

Fig. 2. The Formula of metric evaluation

3. Results and Discussion

3.1. Data Collection

This study employs a cybersecurity attack dataset in CSV format acquired from the Information Systems Bureau of Muhammadiyah University Purwokerto. This dataset provides significant information on suspicious network traffic and potential cybersecurity threats identified by the university's internal security system. It comprises 3000 entries and 25 selected attributes designated as research samples. The selection of this quantity was guided by methodological considerations and prior research, indicating that thousands of data points are adequate to depict the distribution of attack classes while facilitating efficient model training [22]. The data cleansing procedure entailed the elimination of columns considered irrelevant to the research, including 'No.', 'Date', 'URL/Directory', 'Src IP', 'Src Port', 'Rule ID', 'Policy Name', 'Rule Name', 'Impact', 'Description', 'SANGFOR WIKI', and 'Data Packet', as these could potentially lead to data leakage. Furthermore, the columns 'Xff_IP', 'Solution', and 'Reference' were also removed due to their lack of predictive utility for attack classification. Rows exhibiting empty values in the 'Type', 'Dst IP', 'Threat Level', and 'Action' columns were subsequently discarded to maintain dataset integrity. This data cleaning process resulted in a higher-quality dataset that emphasizes relevant predictive features and is prepared for modeling. Subsequently, the Stratified Random Sampling technique was employed to select samples that proportionally represent the ten categories of cyberattacks. [23]. This sampling methodology guarantees that every attack class is precisely represented within the dataset. During execution, the process limits each attack class to a maximum of 848 samples, determined by the class with the highest prevalence. This constraint helps maintain the class distribution ratio, which may affect the performance of the machine learning model [24]. The results of the sampling method demonstrate a distribution consistent with that of the original dataset, with each attack class being represented in accordance with its inherent proportion. This technique has proven effective at preserving the characteristics of each attack class while simultaneously reducing computational complexity [25]. The attributes utilized from the dataset, following data cleaning, are illustrated in Fig. 3.



```
Informasi Dataset:
<class 'pandas.core.frame.DataFrame'>
Index: 2787 entries, 0 to 2999
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   Type          2787 non-null   object
 1   Protocol      2787 non-null   object
 2   Method        2787 non-null   object
 3   Src Zone      2787 non-null   object
 4   Src Location  2787 non-null   object
 5   Src Port      2787 non-null   int64 
 6   Dst Zone      2787 non-null   object
 7   Dst IP        2787 non-null   int64 
 8   Dst Port      2787 non-null   int64 
 9   State Code    2787 non-null   object
 10  Threat Level  2787 non-null   object
 11  Action        2787 non-null   object
dtypes: int64(3), object(9)
memory usage: 283.1+ KB
```

Fig. 3. Dataset Information

3.2. Data Distribution Analysis

This study utilizes the Stratified Random Sampling method by selecting 3000 cyber attack data entries as research samples [22]. This methodology ensures equitable and proportional representation for each attack type within the dataset. Each attack type is limited to a maximum of 848 samples, determined by the class with the highest frequency. This constraint is designed to promote fair representation of each attack type while maintaining a more balanced distribution of data [26]. The implementation of this technique ensures that each data point has an equal probability of selection while preserving the inherent distributional characteristics of each attack category. The class distribution within the research dataset comprises: Information disclosure (848 samples), Website scan (795 samples), Backdoor attack (373 samples), OS command injection (300 samples), Request method filter (185 samples), Web-access cleartext request inspection (129 samples), URL access (82 samples), WebShell Upload (63 samples), File download restriction (8 samples), and Path traversal (4 samples). Subsequently, the dataset was partitioned into training and testing subsets at an 80:20 ratio utilizing the train_test_split function. This partitioning employed the stratify parameter to maintain the class proportions across both subsets. The SMOTE (Synthetic Minority Oversampling Technique) method was applied to the training data to mitigate class imbalance and prevent data leakage. Consequently, the dataset became more representative and was thereby prepared for the subsequent modeling stage.

3.3. Data Exploration

1) Distribution of Attack Types

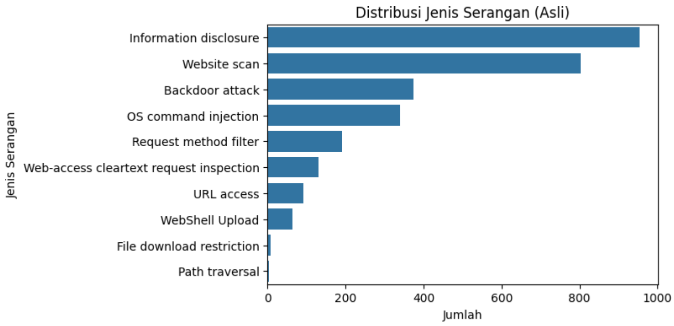


Fig. 4. Attack Types Distribution

The distribution of attack types was analyzed across 3,000 dataset samples using stratified random sampling, yielding 10 attack categories as shown in Fig. 4. This methodology ensures proportional representation of each class, thereby enhancing sample representativeness and facilitating the development of more accurate predictive models. Preliminary analysis indicates an uneven distribution, with Information Disclosure comprising 848 samples, Website Scan with 795 samples, Backdoor Attack with 373 samples, OS Command Injection with 300 samples, Request Method Filter with 185 samples, Web-Access Cleartext Request Inspection with 129 samples, URL Access with 82 samples, WebShell Upload with 63 samples, File Download Restriction with 8 samples, and Path Traversal with 4 samples. This distribution shows dominance of certain classes, while others are underrepresented, potentially affecting the performance of machine learning models. To address this imbalance, the Synthetic Minority Oversampling Technique (SMOTE) was applied to the training data obtained via an 80:20 split. The primary objective of applying SMOTE is to increase the number of samples in minority classes, thereby enabling a more comprehensive analysis of attack patterns.

The imbalance in class distribution in the original dataset reflects the reality of cyber threats in educational institutions, where reconnaissance-based attacks (information disclosure, website scanning) predominate due to their automated, widespread nature. Conversely, advanced attacks such as path traversal and file download restrictions occur sporadically, potentially because perimeter security measures have effectively prevented them or because they necessitate specific knowledge of the target infrastructure. Suppose this dominance of the majority class remains unaddressed. In that case, it may lead the model to develop a bias toward predicting common attacks and to fail to detect more dangerous advanced attacks, despite their lower frequency.

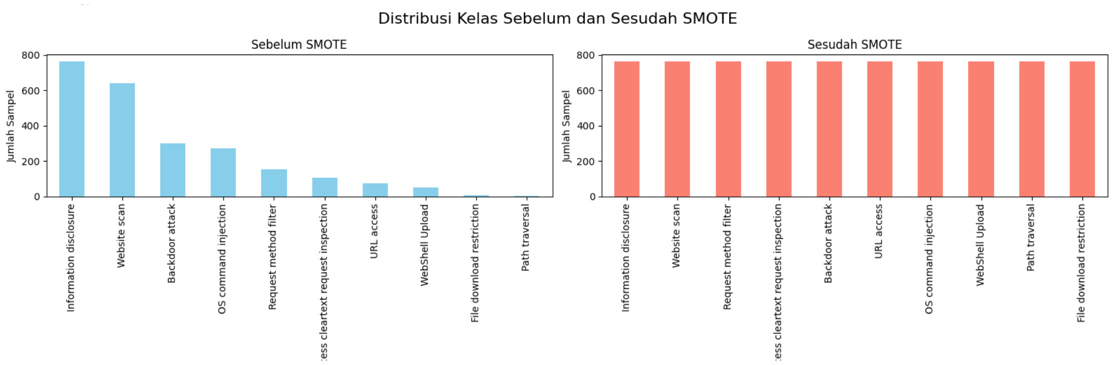


Fig. 5. Attack Types Distribution

Before the application of SMOTE as shown in Fig. 5, the distribution of the training data with an 80% allocation encompassed the following: Information disclosure (678 instances), Website scan (636 instances), Backdoor attack (298 instances), OS command injection (240 instances), Request method filter (148 instances), Web-access cleartext request inspection (103 instances), URL access

(66 instances), WebShell upload (50 instances), File download restriction (7 instances), and Path traversal (3 instances). After implementing SMOTE, each class was balanced to 678 instances, matching the class with the most samples.

2) Feature Importance Distribution Random Forest

An analysis of feature importance utilizing the Random Forest algorithm was conducted to identify the most significant attributes for classifying cyberattacks as shown in Fig. 6. The findings indicate that the 'Threat_Level_Medium' feature has the most important influence, with an importance score of approximately 0.14. This suggests that medium threat levels often correspond to activities that, while not severe, still entail risks, such as website scanning or backdoor probing. Consequently, security systems should focus not only on high-threat activities but also on medium-risk operations that may represent preliminary stages of more extensive cyberattacks. Furthermore, the features 'State_Code_404' and 'Method_GET', which also demonstrate high importance, imply a pattern of HTTP-based attack activity that frequently results in a "Not Found" response, generally due to attempts to access concealed directories or pages. Activities such as these are characteristic of reconnaissance or vulnerability-probing attacks, in which perpetrators seek to identify configuration vulnerabilities. The significance of the 'Action_Deny' feature further indicates that many such activities stem from unauthorized access attempts that the system effectively blocks. In summary, examining feature importance yields valuable insights into each attribute's function and deepens understanding of how the model identifies attacks. This information underpins the formulation of data-driven cybersecurity strategies, prioritizing high- importance attributes as primary targets for early threat detection.

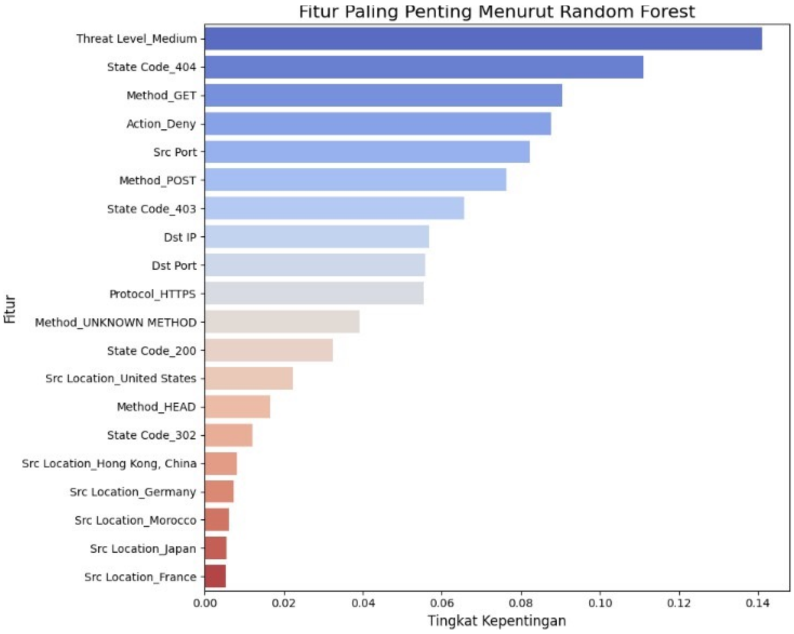
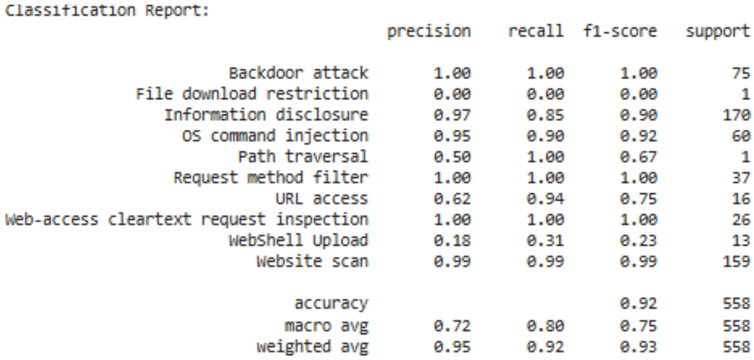


Fig. 6. Random Forest Feature Importance

3.4. Model Performance Evaluation

Based on the test results as shown in Fig. 7, the Random Forest model was evaluated using a classification report and cross-validation to measure its predictive performance. The model achieved an overall accuracy of 92%, with a macro-average F1-score of 0.75 and a weighted-average F1-score of 0.93, demonstrating optimal performance in cyberattack classification.



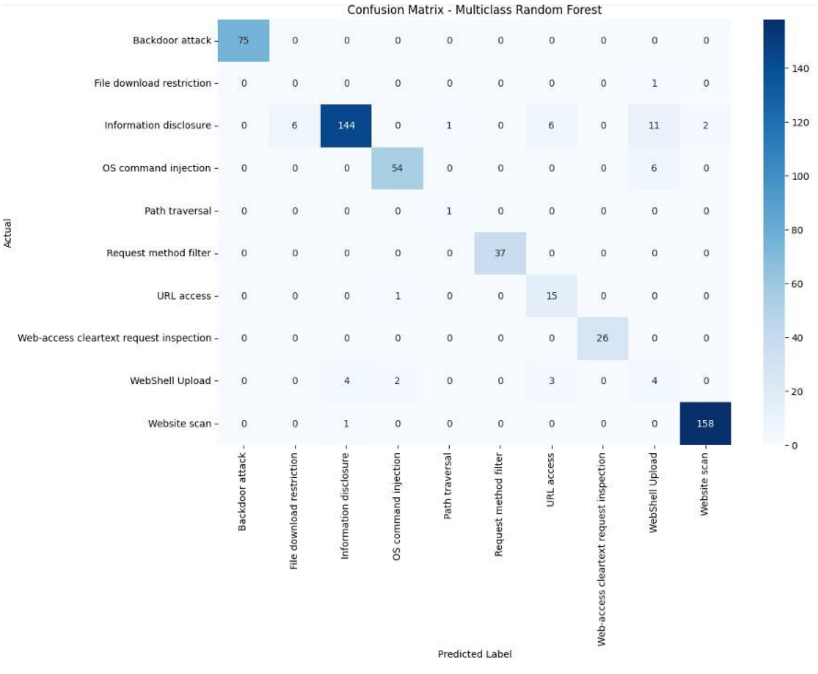
```
Classification Report:
                                      precision    recall  f1-score   support

                   Backdoor attack       1.00      1.00      1.00        75
          File download restriction       0.00      0.00      0.00         1
             Information disclosure       0.97      0.85      0.90       170
               OS command injection       0.95      0.90      0.92        60
                     Path traversal       0.50      1.00      0.67         1
              Request method filter       1.00      1.00      1.00        37
                         URL access       0.62      0.94      0.75        16
Web-access cleartext request inspection       1.00      1.00      1.00        26
                     WebShell Upload       0.18      0.31      0.23        13
                       Website scan       0.99      0.99      0.99       159

                           accuracy                           0.92       558
                          macro avg       0.72      0.80      0.75       558
                       weighted avg       0.95      0.92      0.93       558
```

Fig. 7. Model Evaluation Result

Based on the analysis of the confusion matrix in Fig. 8, the model demonstrated impeccable accuracy in multiple categories, including Backdoor attack, Request method filter, and Web-access cleartext request inspection, each achieving 100% accuracy. The Information disclosure category achieved an accuracy of 84.7%, despite some misclassifications. Such misclassifications, notably within the Information disclosure category, which was frequently confused with WebShell Upload or URL Access, imply similarities in network traffic patterns among these attack types. This phenomenon is presumably attributable to the analogous HTTP request patterns (e.g., GET/POST methods and 403–404 status codes) employed across various attack types. This scenario indicates that the model continues to struggle to differentiate attacks with overlapping technical characteristics, underscoring the potential of behavior-based feature enrichment as a promising avenue for improvement.

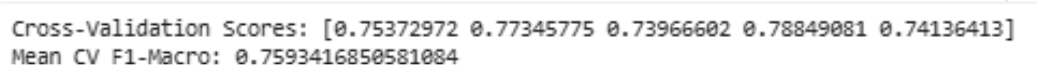


Confusion Matrix - Multiclass Random Forest

Actual	Backdoor attack	File download restriction	Information disclosure	OS command injection	Path traversal	Request method filter	URL access	Web-access cleartext request inspection	WebShell Upload	Website scan
Backdoor attack	75	0	0	0	0	0	0	0	0	0
File download restriction	0	0	0	0	0	0	0	0	1	0
Information disclosure	0	6	144	0	1	0	6	0	11	2
OS command injection	0	0	0	54	0	0	0	0	6	0
Path traversal	0	0	0	0	1	0	0	0	0	0
Request method filter	0	0	0	0	0	37	0	0	0	0
URL access	0	0	0	1	0	0	15	0	0	0
Web-access cleartext request inspection	0	0	0	0	0	0	0	26	0	0
WebShell Upload	0	0	4	2	0	0	3	0	4	0
Website scan	0	0	1	0	0	0	0	0	0	158

Fig. 8. Confusion Matrix Table Result

Further validation employed a 5-fold cross-validation process to assess the model's stability across diverse datasets as shown in Fig. 9. The test outcomes varied from 0.74 to 0.79, with an average F1-Macro score of 0.76. The consistent results across folds indicate that the model performs reliably and is not contingent upon specific data subsets. This stability confirms that integrating the Random Forest algorithm with stratified sampling and SMOTE techniques yields optimal classification performance for cyber attack detection, making it suitable for application as an anomaly detection method in network security systems. Moreover, a comparison between the Random Forest model with and without SMOTE demonstrates that SMOTE elevates the macro-average F1-score from 0.68 to 0.75. This enhancement signifies that the data balancing process effectively improves the model's capacity to classify the minority class without compromising accuracy on the majority class.



```
Cross-Validation Scores: [0.75372972 0.77345775 0.73966602 0.78849081 0.74136413]
Mean CV F1-Macro: 0.7593416850581084
```

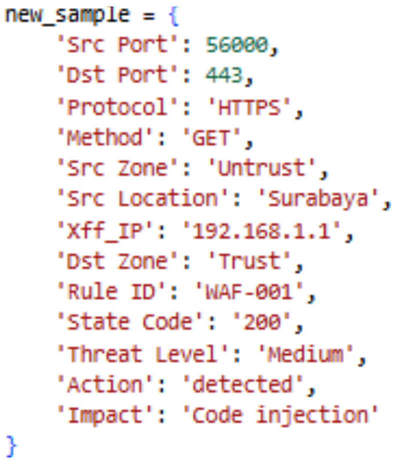
Fig. 9. CV 5-Fold Accuracy

These findings are consistent with numerous prior studies. Alshamy et al. (2021) demonstrated comparable improvement on the NSL-KDD dataset, with the IDS-SMOTE-RF model achieving an F1-score of 99.88% for binary classification. They outperformed other algorithms, including AdaBoost, Logistic Regression, and SVM [27]. In a more specific context, recent research conducted by Kumar and colleagues (2024) reported that the application of Random Forest combined with SMOTE achieved an F1-score of 97.41% and an accuracy of 99.90%, thereby demonstrating highly competitive performance on datasets characterized by more complex attack features [28].

The performance of this research model (F1-score: 0.75; accuracy: 0.89) is competitive with other studies, considering the smaller dataset size (3,000 samples versus >100,000 in the benchmark dataset). The performance gap can be attributed to the smaller dataset, specific attack features in the educational environment, and limited feature variation compared to standard datasets. These results demonstrate that Random Forest-SMOTE remains effective in institutional settings with limited data.

3.5. Detection of Attacks on New Data

At this stage, a new data sample is provided to the model to evaluate how classification outcomes are produced from the available features.



```
new_sample = {
    'Src Port': 56000,
    'Dst Port': 443,
    'Protocol': 'HTTPS',
    'Method': 'GET',
    'Src Zone': 'Untrust',
    'Src Location': 'Surabaya',
    'Xff_IP': '192.168.1.1',
    'Dst Zone': 'Trust',
    'Rule ID': 'WAF-001',
    'State Code': '200',
    'Threat Level': 'Medium',
    'Action': 'detected',
    'Impact': 'Code injection'
}
```

Fig. 10. Parameters New Data

The trained Random Forest model is subsequently evaluated using novel data characterized by attributes such as Src Port = 56000, Dst Port = 443, Protocol = HTTPS, Method = GET, Src Zone = Untrust, Dst Zone = Trust, Threat Level = Medium, and Impact = Code injection. According to this data, the model predicted that the sample belonged to the Website Scan attack category. The results demonstrate that the model effectively classified the new data into one of the attack categories within the dataset. These findings further indicate that the model can accurately interpret new data when combined with numerical and categorical attributes, yielding predictions consistent with prior evaluation outcomes. Precise predictions on new data suggest that the model not only memorizes patterns in the training data but also generalizes to novel attack patterns. This affirms the model's potential as an early-detection system in a dynamic, real-world network environment. The successful classification of new data further reinforces the model's generalization capacity and underscores its potential for effective deployment in identifying genuine cybersecurity threats.

4. Conclusion

This investigation successfully established a cyberattack detection framework utilizing a Random Forest classifier in conjunction with the SMOTE technique. This methodology enhanced the macro-average F1-score from 0.68 to 0.75 and achieved an accuracy rate of 92%. The application of 5-fold cross-validation yielded consistent results, with scores ranging from 74 to 79 and an average of 76, thereby demonstrating robustness and reliable generalization. The study advances three principal contributions: firstly, confirming the efficacy of the Random Forest with SMOTE on a limited dataset of 3,000 samples obtained from a single institution; secondly, emphasizing the significance of monitoring medium-level threats as an early warning mechanism through feature importance analysis; and thirdly, underscoring the necessity of a layered detection strategy to address overlapping attack characteristics. The limitations of the study include reliance on a dataset from a single institution within a constrained timeframe, which limits its applicability across diverse contexts; the absence of temporal features such as request rate and session duration; and an evaluation conducted in a controlled environment, lacking deployment in real-time operational settings. Future research should focus on incorporating temporal and behavioral analytics, exploring ensemble algorithms such as XGBoost, implementing continuous learning to enable adaptive responses to evolving attack patterns, validating the model across datasets from multiple institutions, and developing a real-time visualization dashboard integrated with threat intelligence. In practice, the model could be incorporated into institutional Security Information and Event Management (SIEM) systems through a hybrid approach that combines automated filtering for high-confidence attacks with manual review of uncertain cases. This strategy aims to optimize security resource allocation while maintaining detection accuracy suitable for operational deployment.

References

[1] A. Delplace, S. Hermoso, and K. Anandita, "Cyber Attack Detection thanks to Machine Learning Algorithms," Jan. 2020, [Online]. Available: http://arxiv.org/abs/2001.06309

[2] Badan Siber dan Sandi Negara (BSSN), "Lanskap Keamanan Siber Indonesia 2024," 2024. Accessed: May 26, 2025. [Online]. Available: https://www.bssn.go.id/

[3] K. A. P. da Costa, J. P. Papa, C. O. Lisboa, R. Munoz, and V. H. C. de Albuquerque, "Internet of Things: A survey on machine learning-based intrusion detection approaches," *Computer Networks*, vol. 151, pp. 147–157, Mar. 2019, doi: 10.1016/j.comnet.2019.01.023.

[4] K. Razzaq and M. Shah, "Advancing cybersecurity through machine learning: A scientometric analysis of global research trends and influential contributions," Jun. 2025, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/jcp5020012.

[5] M. Alduailij, Q. W. Khan, M. Tahir, M. Sardaraz, M. Alduailij, and F. Malik, "Machine- Learning-Based DDoS Attack Detection Using Mutual Information and Random Forest Feature Importance Method," *Symmetry (Basel)*, vol. 14, no. 6, Jun. 2022, doi: 10.3390/sym14061095.

[6] "Deteksi serangan siber pada jaringan komputer menggunakan metode random forest," in *Seminar Nasional Teknologi Informasi*, 2024. [Online]. Available: https://bit.ly/CyberSecurityAttacks.

[7] T. Wu, H. Fan, H. Zhu, C. You, H. Zhou, and X. Huang, "Intrusion detection system combined enhanced random forest with SMOTE algorithm," *EURASIP J Adv Signal Process*, vol. 2022, no. 1, Dec. 2022, doi: https://doi.org/10.1186/s13634-022-00871-6.

[8] M. A. Talukder, M. Khalid, and N. Sultana, *A hybrid machine learning model for intrusion detection in wireless sensor networks leveraging data balancing and dimensionality reduction*, vol. 15, no. 1. Nature Publishing Group UK London, 2025, p. 4617. doi: https://doi.org/10.1038/s41598-025-87028-1.

[9] S. Bagui and K. Li, "Resampling imbalanced data for network intrusion detection datasets," *J Big Data*, vol. 8, no. 1, Dec. 2021, doi: 10.1186/s40537-020-00390-x.

[10] A. Abdelkhalek and M. Mashaly, "Addressing the class imbalance problem in network intrusion detection systems using data resampling and deep learning," *Journal of Supercomputing*, vol. 79, no. 10, pp. 10611–10644, Jul. 2023, doi: 10.1007/s11227-023- 05073-x.

[11] M. Soylu and R. Das, "Prediction and graph visualization of cyberattacks using graph attention networks," *Comput Secur*, vol. 157, p. 104534, 2025, doi: https://doi.org/10.1016/j.cose.2025.104534.

[12] C. Fan, M. Chen, X. Wang, J. Wang, and B. Huang, "A Review on Data Preprocessing Techniques Toward Efficient and Reliable Knowledge Discovery From Building Operational Data," Mar. 2021, *Frontiers Media S.A.* doi: 10.3389/fenrg . 2021.652801.

[13] M. P. Pulungan, A. Purnomo, and A. Kurniasih, "Penerapan SMOTE untuk mengatasi imbalance class dalam klasifikasi kepribadian MBTI menggunakan naive bayes classifier," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 11, no. 5, pp. 1033–1042, Oct. 2024, doi: 10.25126/jtiik.2024117989.

[14] G. W. Cha, H. J. Moon, and Y. C. Kim, "Comparison of random forest and gradient boosting machine models for predicting demolition waste based on small datasets and categorical variables," *Int J Environ Res Public Health*, vol. 18, no. 16, Aug. 2021, doi: 10.3390/ijerph18168530.

[15] Y. M. Indah, R. Aristawidya, A. Fitrianto, E. Erfiani, and L. M. R. D. Jumansyah, "Comparison of Random Forest, XGBoost, and LightGBM Methods for the Human Development Index Classification," *Jambura Journal of Mathematics*, vol. 7, no. 1, pp. 14–18, Jan. 2025, doi: 10.37905/jjom.v7i1.28290.

[16] I. Muhamad and M. Matin, "Hyperparameter Tuning menggunakan GridsearchCV pada Random Forest untuk Deteksi Malware," Jurnal Informatika dan Komputer, vol. 8, no. 2, pp. 45–52, 2023.

[17] P. A. doost, S. S. Moghadam, E. Khezri, A. Basem, and M. Trik, "A new intrusion detection method using ensemble classification and feature selection," Sci Rep, vol. 15, no. 1, Dec. 2025, doi: 10.1038/s41598-025-98604-w.

[18] M. K. Suryadewiansyah, T. Endra, and E. Tju, "Jurnal Nasional Teknologi dan Sistem Informasi Naïve Bayes dan Confusion Matrix untuk Efisiensi Analisa Intrusion Detection System Alert", doi: 10.25077/TEKNOSI.v8i2.2022.081-088.

[19] I. Markoulidakis and G. Markoulidakis, "Probabilistic confusion matrix: A novel method for machine learning algorithm generalized performance analysis," Technologies (Basel), vol. 12, no. 7, Jul. 2024, doi: 10.3390/technologies12070113.

[20] J. Brownlee, Machine learning mastery. Machine Learning Mastery, 2022.

[21] E. Dikici, X. Nguyen, N. Takacs, and L. M. Prevedello, "Prediction of Model Generalizability for Unseen Data: Methodology and Case Study in Brain Metastases Detection in T1-Weighted Contrast-Enhanced 3D MRI."

[22] Y. Xie, M. Cheng, Y. Chen, and D. Zhang, "An Internet Intrusion Detection Method Based on Altered Triplet Attention ResNet," in 2025 37th Chinese Control and Decision Conference (CCDC), IEEE, 2025, pp. 2995–3000.

[23] M. A. S. Arifin et al., "Oversampling and undersampling for intrusion detection system in the supervisory control and data acquisition IEC 60870-5-104," IET Cyber-Physical Systems: Theory and Applications, vol. 9, no. 3, pp. 282–292, Sep. 2024, doi: 10.1049/cps2.12085.

[24] M. Mujahid et al., "Data oversampling and imbalanced datasets: an investigation of performance for machine learning and feature engineering," J Big Data, vol. 11, no. 1, Dec. 2024, doi: 10.1186/s40537-024-00943-4.

[25] N. Abedzadeh and M. Jacobs, "A reinforcement learning framework with oversampling and undersampling algorithms for intrusion detection system," Applied Sciences, vol. 13, no. 20, p. 11275, 2023.

[26] W. Chen, K. Yang, Z. Yu, Y. Shi, and C. L. P. Chen, "A survey on imbalanced learning: latest research, applications and future directions," Artif Intell Rev, vol. 57, no. 6, Jun. 2024, doi: 10.1007/s10462-024-10759-6.

[27] R. Alshamy and M. A. Akcayol, "Intrusion Detection Model Using Machine Learning Algorithms on Nsl-Kdd Dataset," International Journal of Computer Networks and Communications, vol. 16, no. 6, pp. 75–88, Nov. 2024, doi: 10.5121/ijcnc.2024.16605.

[28] MD Shadman Soumik, "A comparative analysis of Network Intrusion Detection (NID) using Artificial Intelligence techniques for increase network security," International Journal of Science and Research Archive, vol. 13, no. 2, pp. 4014–4025, Dec. 2024, doi: 10.30574/ijsra.2024.13.2.2664.

AUTHORS BIBLIOGRAPHY

RATNA KOMALASARI was born in Ciamis, West Java, Indonesia, in 2002. She is currently pursuing a Bachelor's degree in Informatics Engineering at the Faculty of Teknik and Science, with primary research interests in data analysis and processing in cybersecurity.

MUKHLIS PRASETYO AJI was born in Purbalingga in 1984. He obtained his bachelor's degree in Electrical Engineering from Muhammadiyah University of Purwokerto. He pursued his master's degree at the Islamic University of Indonesia in the Master of Informatics program with a concentration in digital forensics. He is currently pursuing his doctoral studies at Diponegoro University in the field of Digital Forensics. His current activities include teaching in the Computer Science Department and serving as the Director of the Digital Forensics Center at Muhammadiyah University of Purwokerto. The Digital Forensics Center has been in operation since 2020. Through this center, he has developed the ability to analyze cybercrimes and become an expert in various cases, having resolved 190 cases, analyzed 430 electronic and digital pieces of evidence, and developed the Mobile Cyber Forensics innovation—a mobile laboratory for handling cybercrimes. Through this Digital Forensics Center of Excellence, it will function as a Center of Excellence for Investigation and Education. In addition to being a lecturer, he also serves as the CEO of PT Datatrace Forensics Lab, a digital forensics startup that provides education and consulting services for cybercrime handling.

MAULIDA AYU FITRIANI She obtained her bachelor's degree in Computer Science and subsequently pursued her master's degree in the Same Field. She is currently a lecturer in the Informatics Study Program at Universitas Muhammadiyah Purwokerto. Her areas of expertise include Software Engineering, Artificial Intelligence, and Natural Language Processing (NLP). Her research agenda is interdisciplinary, intersecting with psychology and health. In the academic realm, she teaches courses in Programming, Intelligent Systems, Numerical Computing, NLP, and Research Methodology, and she actively develops structured learning resources, including modules, presentations, question banks, and technical guides. She is also involved in mentoring and delivering AI literacy training for teachers and students.

AGUNG PURWO WICAKSONO was born in Madiun in 1983. He obtained his bachelor's degree in Informatics Engineering from the Islamic University of Indonesia and pursued his master's degree in Computer Science in the Master of Informatics program at the same university. His current activities include teaching in the Informatics Engineering Department at Muhammadiyah University of Purwokerto. Computer networks is the field of study that he specializes in.