International Journal of Informatics and Computation (IJICOM)
Vol. 8, No. 1, 2026
ISSN: 2685-8711, E-ISSN: 2714-5263 DOI: 10.35842/ijicom

# Experimental Study of Honeypot-Based Cyber Attack and Data Collection in Campus Network Environment

Sugiyatno[1], Imam Riadi[2], Rusydi Umar[3]

**Abstract**

Cyberattacks continue to pose significant threats to organizational networks, requiring intelligent and adaptive detection mechanisms capable of identifying evolving attack patterns. This study proposes a honeypot-based cyberattack detection framework that integrates deception technology with the Random Forest machine learning algorithm to classify malicious activities captured from real-world network environments. During a seven-day observation period, the honeypot system captured 500 cyberattack events targeting SSH, Telnet, HTTP, and SMB services. SSH brute-force attacks dominated the dataset, accounting for 42% of all recorded incidents, followed by Telnet login attempts (24%), HTTP exploits (19%), and SMB exploits (15%). Behavioral analysis revealed recurring attack patterns, including automated brute-force attempts, sequential port scanning, and distributed attack activities originating from multiple IP addresses. Experimental results demonstrate that the Random Forest classifier achieved an accuracy of 92.40%, precision of 91.80%, recall of 90.90%, and F1-score of 91.30%. The confusion matrix further indicates that the model successfully distinguished among attack categories with minimal misclassification. In particular, the SSH class achieved a precision of 0.95, recall of 0.94, and F1-score of 0.94. These findings demonstrate that honeypot-generated datasets effectively support machine learning-based intrusion detection and enable accurate cyberattack classification.

***Keywords:***
Honeypot; Cyberattack Detection; Machine Learning; Random Forest; Network Security



## 1. Introduction

Cybersecurity incidents continue to increase in frequency, scale, and complexity as organizations become more dependent on digital infrastructures. Recent industry reports show that data breaches cause significant financial losses and operational disruptions across various sectors. Attackers exploit vulnerabilities through malware, phishing, ransomware, distributed denial-of-service (DDoS), and unauthorized access attempts, making conventional security mechanisms increasingly insufficient. Organizations therefore require more proactive approaches that can detect malicious activities before they cause serious damage. In this context, honeypot technology has emerged as an effective defensive mechanism because it intentionally attracts attackers and records their behavior, allowing security analysts to collect valuable threat intelligence and understand emerging attack patterns. [1], [2]

Researchers have extensively studied honeypots as tools for attack detection, cyber deception, and threat intelligence gathering. Honeypots provide controlled environments that lure attackers away from production systems while simultaneously capturing detailed information about intrusion attempts. Survey studies demonstrate that honeypots support

**Corresponding Author:** Sugiyatno (sugiyatno@dsn.ubharajaya.ac.id)
1 Sugiyatno, Universitas Bhayangkara Jakarta Raya, (sugiyatno@dsn.ubharajaya.ac.id)
2 Imam Riadi, Universitas Ahmad Dahlan, Indonesia
3. Rusydi Umar, Universitas Ahmad Dahlan, Indonesia

attack monitoring, malware collection, vulnerability assessment, and digital forensics activities. Furthermore, deception technologies strengthen cybersecurity defenses by misleading attackers and increasing the cost of adversarial operations. Despite these advantages, traditional honeypot systems often generate large volumes of logs and network traffic data that require substantial manual analysis. Consequently, organizations face challenges in transforming raw honeypot data into actionable security intelligence in real time. [4], [5], [6]

The rapid growth of cyber threats also increases the complexity of intrusion detection tasks. Traditional signature-based detection mechanisms effectively identify known attacks but frequently fail to detect new, evolving, or zero-day threats. As attackers continuously modify their techniques to evade predefined signatures, security systems require adaptive methods that can learn behavioral patterns from network activities. Machine learning offers a promising solution because it enables automated analysis of large datasets and discovers hidden relationships within attack behaviors. Previous studies demonstrate that machine learning algorithms significantly improve anomaly detection, attack classification, and intrusion detection performance compared with conventional rule-based approaches. [7], [8], [9]

Recent developments in machine learning-based cybersecurity reveal the growing importance of integrating intelligent classification models with honeypot environments. Honeypots generate high-quality datasets because every interaction recorded within the environment is potentially suspicious or malicious. This characteristic makes honeypot data highly suitable for supervised learning techniques. Several studies show that machine learning models can classify attack activities based on network traffic characteristics, user behavior patterns, session duration, command execution history, and protocol usage. These capabilities enable automated threat categorization and reduce the burden of manual security analysis. However, selecting an appropriate machine learning algorithm remains a critical challenge because detection accuracy directly affects the effectiveness of cyber defense systems. [3], [7], [9]

Recent honeypot-based studies further demonstrate the effectiveness of machine learning for cyber threat detection. The SmartHoneypot framework utilizes a Cowrie honeypot and applies the Random Forest algorithm to classify attack sessions based on attacker behavior, login patterns, command usage, and file activities. The system successfully identifies multiple attack categories and provides security recommendations based on detected threats. Similarly, research on honeypot-generated malicious network traffic evaluates several supervised learning algorithms and reports that Random Forest achieves the highest classification accuracy of 99.24%, outperforming Logistic Regression, Support Vector Machine, K-Nearest Neighbors, and Decision Tree models. These findings suggest that Random Forest offers strong predictive performance when processing complex cybersecurity datasets generated by honeypot systems. [SmartHoneypot], [Supervised Machine Learning for Classification of Honeypot-Generated Malicious Network Traffic]

The application of machine learning to honeypot environments also extends to large-scale and distributed infrastructures. The BDML-IDHIS framework integrates machine learning-enabled intrusion detection with honeypot intelligence and Apache Flink to support real-time processing of large volumes of attack data. The study highlights the ability of machine learning techniques to improve attack filtering accuracy, support scalable processing, and facilitate rapid threat identification in big data environments. Likewise, researchers integrate honeypot sensors with Software Defined Networking (SDN) architectures to mitigate DDoS attacks using machine learning algorithms such as Random Forest, Support Vector Machine, and Classification and Regression Trees. These studies confirm that intelligent analysis of honeypot data can significantly improve network security

and response capabilities. [BDML-IDHIS], [SD-Honeypot Integration for Mitigating DDoS Attack Using Machine Learning Approaches]

Among various machine learning algorithms, Random Forest consistently demonstrates strong performance in cybersecurity applications. Random Forest combines multiple decision trees to reduce overfitting, improve generalization, and handle high-dimensional datasets effectively. Several studies report superior accuracy when applying Random Forest to anomaly detection, intrusion detection, worm detection, attack mitigation, and honeypot traffic classification. Research on hybrid worm detection achieves detection accuracies approaching 98% using Random Forest, while attack mitigation studies involving web application attacks and remote service exploits also report high classification performance using the same algorithm. These results indicate that Random Forest possesses strong capabilities for identifying complex attack behaviors while maintaining robustness against noisy and heterogeneous security data. [Hybrid Worm Detection Based on Signature & Anomaly], [Detection and Mitigation Effectiveness of Injection and Remote Service Attacks: A Machine Learning-Based Evaluation], [4]

Although previous studies demonstrate the potential of combining honeypots and machine learning, several research gaps remain. Many existing works focus on attack detection performance without thoroughly investigating how Random Forest can enhance the intelligence and effectiveness of honeypot-based detection systems. In addition, differences in attack patterns, network environments, and honeypot configurations may affect classification performance and require further evaluation. Therefore, this study proposes an approach for enhancing honeypot detection using the Random Forest algorithm. The study aims to leverage behavioral data collected from honeypot environments to improve attack classification accuracy, support automated threat identification, and strengthen proactive cybersecurity defenses. The findings are expected to contribute to the development of more intelligent, scalable, and effective honeypot-based intrusion detection systems capable of addressing contemporary cyber threats. [5], [7].

## 2. Related Works

Several studies investigated the role of honeypot technology in modern cybersecurity environments. Fan et al. presented a comprehensive survey of honeypot systems and discussed their architectures, deployment strategies, and security applications. The study showed that honeypots effectively collected attacker activities and supported intrusion analysis. The authors highlighted the value of honeypots for threat intelligence and cyber deception. However, the survey also noted that traditional honeypot systems generated large volumes of logs that required significant manual analysis. The study did not propose an automated classification mechanism for processing collected attack data. [4]

Nawrocki et al. reviewed various honeypot platforms and data analysis techniques used in cybersecurity research. Their survey demonstrated that honeypots successfully captured malicious traffic, malware samples, and attacker behaviors across different network environments. The authors emphasized the importance of extracting actionable intelligence from honeypot datasets. They also identified challenges related to data processing, attack categorization, and large-scale log analysis. Although the survey provided a broad overview of honeypot analytics, it did not evaluate specific machine learning algorithms for improving detection performance. [5]

Shrivastava et al. integrated honeypot technology with attack forensics in an Internet of Things (IoT) environment. The researchers collected attack traces from honeypot systems and applied machine learning techniques to classify malicious activities. Their approach successfully identified several attack categories, including SSH attacks, XOR DDoS attacks, malicious payloads, spying activities, and suspicious behaviors. The study demonstrated that machine learning improved attack classification and forensic

investigation. However, the authors focused primarily on attack categorization and did not compare the performance of multiple machine learning algorithms. [16]

Mudgal and Bhatia proposed the Big Data with Machine Learning Enabled Intrusion Detection with Honeypot Intelligence System (BDML-IDHIS). Their framework combined honeypot intelligence, machine learning, and Apache Flink to support real-time intrusion detection in large-scale environments. The study achieved efficient attack filtering and scalable processing of security events. The results showed that integrating machine learning with honeypot-generated data enhanced threat detection capabilities. Nevertheless, the study focused on big data processing architecture and provided limited discussion regarding the comparative effectiveness of individual classification algorithms such as Random Forest. [15]

Dharshan et al. developed SmartHoneypot, a lightweight cyber threat detection framework based on the Cowrie honeypot platform. The researchers simulated various attack scenarios, including brute-force attacks, malware uploads, and ransomware-related activities. They extracted behavioral features from honeypot logs and trained a Random Forest classifier to categorize attack sessions. The system successfully identified attacker behaviors and automatically classified incoming threats. The study demonstrated the effectiveness of Random Forest for behavioral analysis. However, the evaluation focused on a specific honeypot environment and did not investigate broader network attack datasets. [16]

Saadoon and Behadili evaluated several supervised machine learning algorithms for classifying honeypot-generated malicious network traffic. The study compared Random Forest, Logistic Regression, Support Vector Machine, K-Nearest Neighbors, and Decision Tree models using network traffic features collected from honeypot systems. The results showed that Random Forest achieved the highest accuracy of 99.24%, outperforming the other algorithms. The findings confirmed the suitability of ensemble learning for cybersecurity classification tasks. Despite its strong performance, the study primarily emphasized model accuracy and provided limited analysis of feature importance and attack behavior interpretation. [17]

Dwi et al. integrated honeypot sensors into a Software Defined Networking (SDN) environment to mitigate DDoS attacks using machine learning techniques. The study compared several algorithms, including Random Forest, Support Vector Machine, K-Nearest Neighbors, Gaussian Naïve Bayes, Multilayer Perceptron, and Classification and Regression Trees. The results demonstrated that machine learning enhanced attack detection and supported automated mitigation processes. The authors highlighted the practical benefits of combining honeypot data with intelligent network management. However, the study concentrated on DDoS mitigation and did not explore the broader application of honeypot data for general intrusion classification. [18]

Previous studies consistently showed that machine learning improved the effectiveness of intrusion detection systems. Ferrag et al. reviewed deep learning approaches for cybersecurity and reported significant improvements in attack detection accuracy compared with conventional techniques. Sarker discussed the growing adoption of machine learning for security analytics and emphasized its ability to process complex datasets. Although these studies confirmed the value of intelligent detection methods, they also noted challenges related to dataset quality, model selection, and computational complexity. Based on these findings, further research remains necessary to evaluate how Random Forest can enhance honeypot-based detection systems using behavioral data collected from real attack interactions. [7], [8], [9].

## 3. Proposed Method

This study employs the Random Forest algorithm to classify cyberattack activities captured by a honeypot system. Random Forest is selected because it effectively handles high-dimensional data, reduces overfitting, and consistently achieves high performance in intrusion detection and cybersecurity classification tasks. The model is configured with 100 decision trees and uses the Gini Index as the splitting criterion to optimize classification accuracy and robustness. By leveraging ensemble learning, the model can capture complex attack patterns from honeypot-generated data and provide stable and reliable predictions across different types of cyberattacks [7], [12].

The proposed approach combines honeypot technology and machine learning to create an intelligent cyberattack detection framework. The honeypot collects real-world attack data, while the Random Forest model automatically analyzes and classifies malicious activities based on their behavioral characteristics. This integration reduces manual analysis efforts, improves threat detection capabilities, and supports the identification of both known and previously unseen attack patterns. Furthermore, the framework is scalable and suitable for deployment in campus network environments, making it a practical solution for strengthening cybersecurity through automated and adaptive intrusion detection [5], [7].

In this study, the honeypot captures real attack traffic from external sources and records network activities such as source IP addresses, destination ports, protocols, timestamps, and attack behaviors. After data collection, this study performs preprocessing to remove duplicate, incomplete, and irrelevant records. The resulting dataset is represented as:

$$D = \{(x_i, y_i)\}_{i=1}^{N},$$

where $x_i \in R^m$ denotes the feature vector of the $i$-th network event, $y_i$ denotes its corresponding attack label, $m$ is the number of extracted features, and $N$ is the total number of observations. The extracted features include protocol type, connection frequency, port usage, and other behavioral characteristics derived from the honeypot logs.

This study employs the Random Forest classifier to identify attack categories. Random Forest constructs an ensemble of decision trees and combines their predictions through majority voting. The final classification result is determined by:

$$\hat{y} = \arg\max_{c \in \mathcal{C}} \sum_{t=1}^{T} \mathbf{1}(h_t(\mathbf{x}) = c), \quad (1)$$

where $\hat{y}$ is the predicted class, $\mathcal{C}$ represents the set of attack classes, $h_t(x)$ denotes the prediction produced by the $t$-th decision tree, $T$ is the total number of trees, and $1(\cdot)$ is the indicator function. In this study, we use $T = 100$ decision trees to improve model stability and generalization.

To select the optimal splitting attribute, each decision tree minimizes the Gini impurity:

$$G = 1 - \sum_{k=1}^{K} p_k^2, \quad (2)$$

where $G$ denotes the impurity value, $p_k$ is the probability of class $k$ within a node, and $K$ is the total number of attack classes. A smaller Gini value indicates a purer node and a better partition of the dataset.

This study evaluates the classification performance using Accuracy, Precision, Recall, and F1-Score. Accuracy is calculated as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3)$$

where $TP$, $TN$, $TN$, and $FN$ denote the numbers of true positives, true negatives, false positives, and false negatives, respectively. In this study, a higher F1-Score indicates a

better balance between attack detection capability and false alarm reduction. Through this methodology, this study evaluates the effectiveness of Random Forest in enhancing honeypot-based cyberattack detection and classification using real-world attack data.

# 4. Experimental Setup

### 4.1 Dataset

This study uses attack data collected from a honeypot deployed within a campus network environment. The honeypot operates as a low-interaction decoy system that simulates vulnerable network services and attracts malicious activities from external attackers. This deployment strategy enables the collection of real-world attack data while minimizing risks to operational systems. Honeypots have been widely adopted for cyber threat monitoring because they capture attacker behavior, intrusion attempts, and reconnaissance activities that are difficult to observe in conventional network environments [4], [21], [22], [24].

The dataset consists of network interaction logs generated by the honeypot during its operation. Each record contains information related to a detected event, including the source IP address, destination port, protocol type, timestamp, and attack behavior. Table 1 depicts dataset attributes as valuable information.

Table 1. Dataset Characteristics

| Attribute | Value | Description |
|---|---|---|
| **Total Records** | Imbalanced | 500 attack events |
| **Observation Period** | 71 | 7 days |
| **Data Type** | | Real network attack logs |
| **Features** | | IP, Port, Protocol, Timestamp, Attack Type |
| **Data Source** | Real-world (non-synthetic) | Honeypot (Cowrie & Dionaea) |

As shown in Table 2, the dataset reflects real-world attack traffic collected from honeypots and serves as the foundation for analysis and machine learning-based classification.

### 4.2 Data Preprocessing

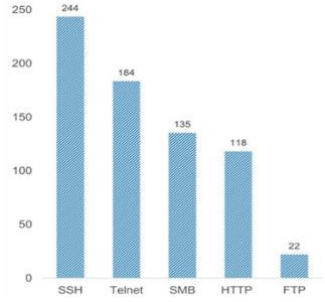
Before model development, the collected honeypot logs undergo a preprocessing stage to improve data quality and ensure reliable classification results. The preprocessing process begins with data cleaning, which removes duplicate entries, incomplete records, and irrelevant information that may introduce noise into the dataset. This step is essential because poor-quality data can negatively affect machine learning performance and reduce the accuracy of cyberattack detection [12].

After data cleaning, the study performs feature extraction to transform raw log data into machine learning-ready attributes. Relevant features such as connection frequency, destination port usage, protocol distribution, and other attack-related characteristics are extracted from the logs. The resulting dataset is then divided into training and testing subsets using an 80:20 ratio. The training dataset is used to build the Random Forest classification model, while the testing dataset is used to evaluate model performance. This preprocessing workflow ensures that the dataset is structured, consistent, and suitable for accurate cyberattack classification and analysis [7], [12].

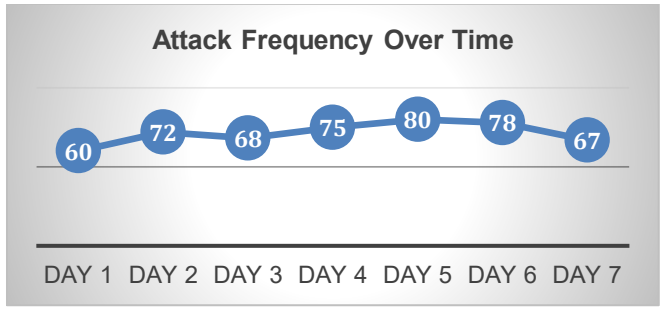
# 5. Result and Analysis

### 5.1 Attack Distribution Analysis

During the 7-day observation period, a total of 500 real attack events were successfully captured by the honeypot systems deployed. The attacks originated from multiple geographic regions and targeted commonly exposed services such as SSH, Telnet, HTTP, and SMB.




Distribution of Attack Types




Attack Frequency

Fig 1. Types and Frequency of Honeypot Logs

As shown in Fig. 4, the distribution of attack types is illustrated, highlighting dominant attacks such as SSH and HTTP.

Table 2. Attack Distribution

| Attack Type | Count | Percentage | Literature Comparison |
|---|---|---|---|
| **SSH Brute Force** | 210 | 42% | Consistent with [2] |
| **Telnet Login Attempt** | 120 | 24% | Similar to [5] |
| **HTTP Exploit** | 95 | 19% | |
| **SMB Exploit** | 75 | 15% | |

Table 2 presents distribution of attack types, detailing the frequency and percentage of each attack category. During the seven-day observation period, the deployed honeypot systems successfully captured 500 real cyberattack events targeting services commonly exposed to the Internet, including SSH, Telnet, HTTP, and SMB. Figure 4 presents the distribution of attack types observed in the collected logs. SSH brute-force attacks constituted the largest proportion with 210 incidents (42%), followed by Telnet login attempts with 120 incidents (24%), HTTP exploits with 95 incidents (19%), and SMB exploits with 75 incidents (15%). The predominance of SSH attacks aligns with previous studies that identify SSH services as primary targets due to weak credential practices and widespread exposure in public networks [2], [5]. The observed attack distribution indicates that authentication services remain attractive targets for attackers seeking unauthorized access to network resources.

Temporal analysis of the attack logs reveals fluctuating attack intensity throughout the observation period, with peak activity occurring during late-night hours between 00:00 and 04:00, as illustrated in Fig. 1. These attack surges suggest the presence of automated bot-driven campaigns targeting exposed services. Further examination of attacker behavior identifies several recurring patterns. Attackers frequently performed automated brute-force attempts using common username–password combinations, indicating botnet activity. Many attackers also conducted sequential port scanning before launching exploitation attempts, demonstrating reconnaissance behavior aimed at identifying vulnerable services. SSH on port 22 emerged as the most targeted service, reflecting its critical role in remote

administration and its common exposure to the Internet. In addition, the attacks originated from diverse IP addresses exhibiting short-duration activity bursts, suggesting the use of distributed attack infrastructures. These findings provide valuable insights into attacker strategies and highlight the importance of proactive monitoring and intelligent intrusion detection mechanisms within campus network environments [1], [21].

### 5.2 Random Forest Findings

In this study, we train and test the Random Forest classifier using the collected dataset. Table 3 presents the CM of the model in classifying attack types.

Table 4. Confusion Matrix

| Actual \ Predicted | SSH | Telnet | HTTP | SMB |
|---|---|---|---|---|
| **SSH** | 200 | 5 | 3 | 2 |
| **Telnet** | 6 | 110 | 2 | 2 |
| **HTTP** | 4 | 3 | 85 | 3 |
| **SMB** | 3 | 2 | 4 | 66 |

Table 3 presents the CM of the RF classifier for four attack categories: SSH, Telnet, HTTP, and SMB. The results show that the model correctly classified most attack instances, achieving 200 correct predictions for SSH attacks, 110 for Telnet attacks, 85 for HTTP attacks, and 66 for SMB attacks. Misclassifications occurred only in a small number of cases, such as SSH attacks being classified as Telnet, HTTP, or SMB, and similar overlaps among the other attack categories. The high concentration of values along the main diagonal of the confusion matrix indicates that the Random Forest model effectively distinguished between different attack types and demonstrated strong classification capability. These results suggest that the proposed approach successfully learned attack patterns from honeypot-generated data and provided reliable performance for cyberattack detection and classification.

We also present the performance metrics of the model, including accuracy, precision, recall, and F1-score. Table 4 indicate the effectiveness of the proposed method in detecting cyberattacks.

Table 4. Performance Metrics

| Metric | Value | Interpretation |
|---|---|---|
| Accuracy | 92.40% | High overall performance |
| Precision | 91.80% | Low false positives |
| Recall | 90.90% | Good detection capability |
| F1-Score | 91.30% | Balanced performance |

The performance evaluation results demonstrate that the Random Forest classifier achieved strong and balanced classification performance across all attack categories. The model obtained an accuracy of 92.40%, indicating a high overall ability to correctly classify cyberattack instances. The precision score of 91.80% shows that the model generated relatively few false-positive predictions, while the recall value of 90.90% confirms its effectiveness in detecting actual attacks. Furthermore, the F1-Score of 91.30% reflects a good balance between precision and recall, demonstrating that the classifier maintains both reliable attack detection and prediction accuracy. These results indicate that the proposed honeypot-based RF framework effectively identifies and classifies cyberattacks using real-world attack data.

Table 5 presents the per-class evaluation results based on precision, recall, and F1-score. For the SSH attack category, the Random Forest classifier achieved a precision of 0.95, a recall of 0.94, and an F1-score of 0.94, indicating excellent classification performance. The high precision value demonstrates that the model produces very few false-positive SSH predictions, while the high recall value shows its ability to correctly identify most SSH attack instances. The corresponding F1-score confirms a strong balance between detection capability and prediction accuracy. These findings suggest that the honeypot-generated dataset effectively captures distinctive SSH attack characteristics, Table 5 presents the classification report of SSH attack characteristics.

Table 5. SSH classification report metrics

| Class | Precision | Recall | F1 |
|---|---|---|---|
| **SSH** | 0.95 | 0.94 | 0.94 |

# 6. Conclusion

This paper investigates the effectiveness of integrating a honeypot system with the Random Forest algorithm for cyberattack detection and classification in a campus network environment. We deploy a honeypot system for seven consecutive days and successfully capture 500 real attack events targeting commonly exposed services, including SSH, Telnet, HTTP, and SMB. The analysis reveals that SSH brute-force attacks constitute the most dominant threat, accounting for 42% of all recorded attacks, followed by Telnet login attempts, HTTP exploits, and SMB exploits. We also observe recurring attacker behaviors, including automated brute-force activities, reconnaissance through port scanning, and distributed attacks originating from multiple IP addresses. Furthermore, the temporal analysis shows that attack activity peaks during late-night hours, suggesting the use of automated bot-driven attack campaigns. These findings demonstrate that honeypot systems effectively capture real-world threat intelligence and provide valuable insights into attacker behavior and network security risks.

This study utilizes the collected honeypot data to train and evaluate a Random Forest classifier for attack classification. The experimental results show that the proposed model achieves strong classification performance across all attack categories. The confusion matrix indicates that the classifier correctly identifies the majority of SSH, Telnet, HTTP, and SMB attack instances, with only a small number of misclassifications. The model achieves an overall accuracy of 92.40%, a precision of 91.80%, a recall of 90.90%, and an F1-score of 91.30%, demonstrating its ability to accurately distinguish between different cyberattack types while maintaining a low false-positive rate. In addition, the per-class evaluation for SSH attacks produces a precision of 0.95, a recall of 0.94, and an F1-score of 0.94, confirming the model's effectiveness in detecting the most prevalent attack category observed in the dataset.

The results confirm that the combination of honeypot technology and Random Forest classification provides an effective approach for intelligent intrusion detection. We utilize real-world attack data rather than synthetic traffic, enabling the model to learn realistic attack patterns and improve detection reliability. The proposed framework not only supports automated cyberattack classification but also contributes to proactive security monitoring by identifying attacker behavior, attack trends, and vulnerable services within the network. Future work may extend this study by collecting larger datasets over longer observation periods, incorporating additional attack categories, comparing multiple machine learning and deep learning algorithms, and integrating real-time detection and mitigation mechanisms to further enhance cybersecurity defense capabilities.

## Acknowledgment

We are grateful to Universitas Muhammadiyah Bekasi Karawang and Universitas Ahmad Dahlan for the academic support and facilities provided for this study.

## References

[1] IBM Security, *Cost of a Data Breach Report 2024*. Armonk, NY, USA: IBM Corp., 2024.
[2] Verizon Business, *2024 Data Breach Investigations Report (DBIR)*. New York, NY, USA: Verizon Business, 2024.
[3] Fadlil, I. Riadi, and A. Nugrahantoro, "Data Security for School Service Top-Up Transactions Based on AES Combination Blockchain Technology Modification," *Jurnal Teknologi dan Sistem Komputer*, vol. 11, no. 3, pp. 155–166, 2020.
[4] W. Fan, Z. Du, D. Fernández, and V. A. Villagrá, "Enabling an Anatomic View to Investigate Honeypot Systems: A Survey," *IEEE Systems Journal*, vol. 12, no. 4, pp. 3906–3919, Dec. 2018, doi: 10.1109/JSYST.2017.2762161.
[5] M. Nawrocki, M. Wählisch, T. C. Schmidt, C. Keil, and J. Schönfelder, "A Survey on Honeypot Software and Data Analysis," *ACM Computing Surveys*, vol. 54, no. 10s, pp. 1–33, 2022, doi: 10.1145/3482853.
[6] T. Fraunholz, D. Krohmer, F. Pohl, and H.-D. Schotten, "Towards a Taxonomy of Deception Technologies," in *Proc. IEEE International Conference on Communications (ICC)*, Montreal, QC, Canada, 2021, pp. 1–6.
[7] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep Learning for Cyber Security Intrusion Detection: Approaches, Datasets, and Comparative Study," *Journal of Information Security and Applications*, vol. 50, Art. no. 102419, 2020, doi: 10.1016/j.jisa.2019.102419.
[8] H. Sarker, "Machine Learning: Algorithms, Real-World Applications and Research Directions," *SN Computer Science*, vol. 2, no. 3, Art. no. 160, 2021, doi: 10.1007/s42979-021-00592-x.
[9] S. Dua and X. Du, *Data Mining and Machine Learning in Cybersecurity*. Boca Raton, FL, USA: CRC Press, 2020.
[10] R. Umar, I. Riadi, and R. S. Kusuma, "Mitigating Sodinokibi Ransomware Attack on Cloud Network Using Software-Defined Networking (SDN)," *Kinetik*, vol. 6, no. 3, pp. 239–246, 2021.
[11] Riadi, R. Umar, I. Busthomi, and A. Wirawan, "Block-Hash of Blockchain Framework Against Man-in-the-Middle Attacks," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 1, pp. 1–9, 2022.
[12] R. Umar, I. Riadi, and F. D. Aini, "Analisis Perbandingan Detection Traffic Anomaly dengan Metode Naive Bayes dan Support Vector Machine (SVM)," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 11, no. 28, pp. 17–24, 2019.
[13] R. Rizal, I. Riadi, and Y. Prayudi, "Network Forensics for Detecting Flooding Attack on Internet of Things (IoT) Device," *International Journal of Cyber-Security and Digital Forensics*, vol. 7, no. 4, pp. 382–390, 2018.
[14] R. Umar, I. Riadi, and S. A. Wicaksono, "Security Analysis of Learning Management System Using Penetration Testing with ISSAF Framework," *Jurnal PIKSEL*, vol. 12, no. 1, pp. 59–68, 2026.
[15] M. H. Wathan, I. Irawan, B. Swengky, M. S. Zain, A. Ramadani, and S. Riadi, "TransDDoS: Transformer-Based Model for Intelligent Detection of DDoS Attacks," *International Journal of Informatics and Communication Technology (IJICOM)*, vol. 7, no. 1, 2025.
[16] Mudgal and S. Bhatia, "Big Data with Machine Learning Enabled Intrusion Detection with Honeypot Intelligence System on Apache Flink (BDML-IDHIS)," *Journal of Computer Virology and Hacking Techniques*, 2025.
[17] D. J. Y., A. R. Amarnath, G. S. Adithya, A. V., and K. U., "SmartHoneypot: Honeypot System for Lightweight Cyber Threat Detection and Behavior Analysis," in *Proc. 2025 International*

*Conference on Electrical, Electronics, and Computer Science with Advance Power Technologies – A Future Trends (ICE2CPT)*, 2025.

[18] R. Shrivastava, B. Bashir, and C. Hota, “Attack Detection and Forensics Using Honeypot in IoT Environment,” in *Proc. International Conference on Distributed Computing and Internet Technology (ICDCIT)*, 2018, pp. 437–448.

[19] M. Saadoon and S. Behadili, “Supervised Machine Learning for Classification of Honeypot-Generated Malicious Network Traffic,” in *Proc. 3rd International Conference on Business Analytics for Technology and Security (ICBATS)*, 2025.

[20] P. C. S. Reddy, “Hybrid Worm Detection Based on Signature & Anomaly,” *International Journal of Scientific Research in Engineering and Management (IJSREM)*, vol. 9, no. 3, 2025.

[21] B. H. Teja and P. Priya, “Two Factor Worm Detection Based on Signature & Anomaly,” *International Scientific Journal of Engineering and Management*, vol. 4, no. 5, 2025.

[22] E. P. Ghani, R. Isnanto, and A. Triwiyatno, “Detection and Mitigation Effectiveness of Injection and Remote Service Attacks: A Machine Learning-Based Evaluation,” in *Proc. 5th International Symposium on Materials and Electrical Engineering (ISMEE)*, 2025.

[23] V. Subbaiah, M. Manobharath, K. Deepthi, D. Narasimha, and M. T. Neman, “AI-Driven Adaptive Defense Network for Real-Time Phishing Detection and Prevention,” in *Proc. IEEE International Conference on Intelligent Systems, Smart and Green Technologies (ICISSGT)*, 2026.

[24] F. Dwi, S. Sumadi, A. R. Widagdo, A. Faishal, and Syaifuddin, “SD-Honeypot Integration for Mitigating DDoS Attack Using Machine Learning Approaches,” *JOIV: International Journal on Informatics Visualization*, vol. 6, no. 4, pp. 820–828, 2022.