

A Hybrid Random Forest–Support Vector Machine Model for Classifying Cyberattacks Using SSH Honeypot Log Data

Firrofiquil A'la Kadram¹, Muhammad Faisal^{2*}, Runal Rezkiawan³, Aedah Abd Rahman⁴, Muhammad Syafaat S.Kuba⁵, Andi Makbul Syamsuri⁶

Abstract—Secure Shell (SSH) is widely used for remote server administration; however, its exposure to public networks makes it a frequent target of cyberattack activities. This article presents an attack classification framework for Cowrie SSH Honeypot logs, using a hybrid Random Forest–Support Vector Machine approach. The raw logs were processed through log auditing, data cleaning, session reconstruction, class labeling, feature construction, feature selection, and model evaluation. The dataset consisted of 38 log files, 213,494 valid records, 29,099 unique sessions, and 1,618 unique source IP addresses. The analyzed attack categories included failed authentication, other suspicious interactions, successful compromise, and command execution. Random Forest ranked ten candidate behavioral features, selected the five highest-ranked features within each rolling fold, and generated time-aware out-of-fold class-probability estimates. The selected features and RF probability outputs were combined, standardized, and passed to a support vector machine with a radial basis function (RBF) kernel as the stage-1 classifier; sessions predicted as successful interactions were then refined into successful compromise or command execution based on observed command activity. Evaluation using eight rolling time-based folds achieved an accuracy of $99.65\% \pm 0.38\%$, a balanced accuracy of $97.02\% \pm 5.77\%$, and a macro F1-score of $97.84\% \pm 4.37\%$ for the final four-class classification. The event_count, username_diversity, interarrival_std, and password_diversity features were selected in all folds. Theoretically, RF–SVM provides a stable session-level representation; in practice, it

supports honeypot log triage and the prioritization of higher-risk SSH sessions.

Index Terms—SSH Honeypot, Cowrie, attack classification, random forest, support vector machine.

I. INTRODUCTION

Secure Shell (SSH) is an essential service widely used for remote server administration. However, because this service is often exposed to public networks, SSH has become one of the main targets of cyberattacks. Exposed services may be subjected to brute-force attempts, exploration of username and password combinations, unauthorized access, and follow-up activities after successful authentication [1], [2]. Attacks on SSH do not only represent attempts to access a system, but also generate behavioral traces that can be analyzed, such as login attempt frequency, credential variation, session duration, inter-event time patterns, and commands executed by attackers. These traces are important because they provide insight into attack intensity, access-attempt patterns, and the likelihood of further activity after an attacker enters the observed environment. Therefore, SSH attack analysis needs to consider not only whether access attempts succeed or fail, but also how the attacker's behavior evolves throughout a session.

An SSH honeypot is used to observe and record attacker activity targeting SSH services. Unlike production systems that provide real operational services, honeypots are deployed as controlled environments intended to attract attacker interactions and capture their behavior in detail. Adaptive honeypots can support the identification of attack behavior patterns, while cyber deception approaches can redirect attacker activities to prepared environments for observation [3], [4]. In addition, honeypots can provide information about attacker behavior with a low false detection rate [5]. Moving-target defense also positions honeypots as part of a more adaptive, deception-based strategy to address cyber threats [6]. Through this approach, attacker activities can be observed without directly involving production systems, making the collected data more suitable for attack behavior analysis.

Received: 8 June 2026; Revised: 14 July 2026; Accepted: 26 July 2026.

*Corresponding author

¹Firrofiquil A'la Kadram, Department of Informatics, Universitas Muhammadiyah Makassar (e-mail: 105841115023@student.unismuh.ac.id).

²Muhammad Faisal, Department of Informatics, Universitas Muhammadiyah Makassar (e-mail: muhfaisal@unismuh.ac.id).

³Runal Rezkiawan, Department of Informatics, Universitas Muhammadiyah Makassar (e-mail: runal@unismuh.ac.id).

⁴Aedah Abd Rahman, School of Science and Technology, Asia e University (e-mail: aedah.abdrahman@acu.edu.my).

⁵Andi Makbul Syamsuri, Department of Water Resources Engineering, Universitas Muhammadiyah Makassar (e-mail: amakbulsyamsuri@unismuh.ac.id).

⁶Muhammad Syafaat S. Kuba, Department of Water Resources Engineering, Universitas Muhammadiyah Makassar (e-mail: syafaat_skuba@unismuh.ac.id).

Cowrie Honeypot is one of the most widely used SSH honeypots for recording attacker activity. This system can capture authentication attempts, connection information, client identities, session activities, and command execution. The generated logs not only serve as technical records but also provide a basis for understanding attackers' operational patterns and for developing behavior-based security analysis. Behavior-based security datasets are important because they can capture attack activities in a more detailed and structured manner [7]. In the SSH context, interactive honeypot logs can capture connection attempts, brute-force authentication attempts, and command execution, making them relevant for analyzing session-based attack behavior [8]. Therefore, Cowrie logs have analytical value because they can connect authentication activities, session interactions, and attacker commands into a more complete behavioral sequence.

The use of honeypot logs for attack classification still presents several challenges. The generated logs are generally recorded at the event level, arranged sequentially, produced in diverse formats, and contain attributes that are not always relevant to the classification task. Realistic and structured cybersecurity datasets are important for supporting data-driven analysis, while temporal changes in patterns also pose a challenge for cybersecurity analysis [9], [10]. In addition, changes in data patterns, concept drift, and feature drift need to be considered when an IDS operates in dynamic environments [11]. If raw logs are directly used, the model may receive repetitive, less informative, or overly rule-specific features. This condition can increase computational complexity and reduce the model's ability to recognize more general attack patterns. Therefore, log data need to be reconstructed into session-based representations so that each attacker interaction can be described through more concise and meaningful features. This representation is also important because attack behavior is often reflected by a sequence of activities rather than by a single log event.

The hybrid Random Forest-Support Vector Machine approach was used to develop an attack classification model based on SSH honeypot logs. In this workflow, Random Forest was employed to select features and identify behavioral attributes with the strongest contribution, while SVM was used as the final classifier. Feature selection is important for reducing data dimensionality and retaining more informative attributes [12]. This model combination was selected to address attack patterns that are not always linear, imbalanced class distributions, and the need for features that accurately represent session behavior. Cowrie log data were processed through log auditing, data cleaning, session reconstruction, class labeling, feature construction, feature selection, and model evaluation. The analyzed attack classes included failed authentication, other suspicious interactions, successful compromise, and command execution. Through this workflow, the classification process is expected to capture not only the final status of an interaction, but also the behavioral characteristics that appear during the session.

The main contributions of this article are presented in three

aspects. First, Cowrie SSH Honeypot logs are reconstructed into session-based representations to enable a more structured analysis of attacker behavior. Second, the hybrid Random Forest-SVM approach integrates feature selection and classification into a single modeling workflow. Third, the evaluation is conducted using a rolling time-based fold scheme, so that model performance can be assessed in chronological order, which is better suited to the dynamic nature of honeypot attack activities. Through this approach, model performance is evaluated not only by accuracy but also by balanced accuracy, macro F1-score, misclassification patterns, and feature stability. These contributions show that this article emphasizes not only classification performance, but also session representation and feature interpretation relevant to SSH attack analysis.

II. RELATED WORK

Machine learning approaches have been widely adopted in cybersecurity to support intrusion detection, anomaly detection, and attack classification. Artificial intelligence contributes to threat detection and analysis, while machine learning models have also been applied to IoT-based attack detection [1], [13]. Random Forest is suitable for handling complex data, reducing noise effects, and identifying feature contributions, whereas Support Vector Machine (SVM) can construct decision boundaries in high-dimensional feature spaces. This combination is relevant for SSH honeypot logs because attack patterns are not always linear, class distributions are imbalanced, and the classification process requires representative behavioral features. Machine learning and deep learning models have also been compared in data-driven classification tasks, showing that model selection can affect classification performance [14]. SVM-based classification has been applied with feature extraction and optimization techniques to improve accuracy [15]. In addition, hybrid machine learning frameworks have been used to support decision-making by combining classification models with optimization methods [16].

Random Forest, SVM, ensemble learning, semi-supervised learning, and temporal models have been applied in various intrusion detection and cyberattack analysis scenarios. Context-based and ensemble approaches can support threat detection and intrusion detection processes [17], [18]. Semi-supervised learning can improve IDS detection capability [19]. However, most existing approaches still focus on general network traffic, benchmark IDS datasets, IoT environments, or broad anomaly detection tasks. Class imbalance also remains a challenge in multi-class IDS because it can affect model performance and interpretation [20].

III. RESEARCH METHOD

The method was designed to convert raw Cowrie SSH Honeypot logs into a session-based dataset suitable for attack classification. The workflow includes log auditing, data cleaning, session reconstruction, class labeling, feature construction, feature selection, and classification using the

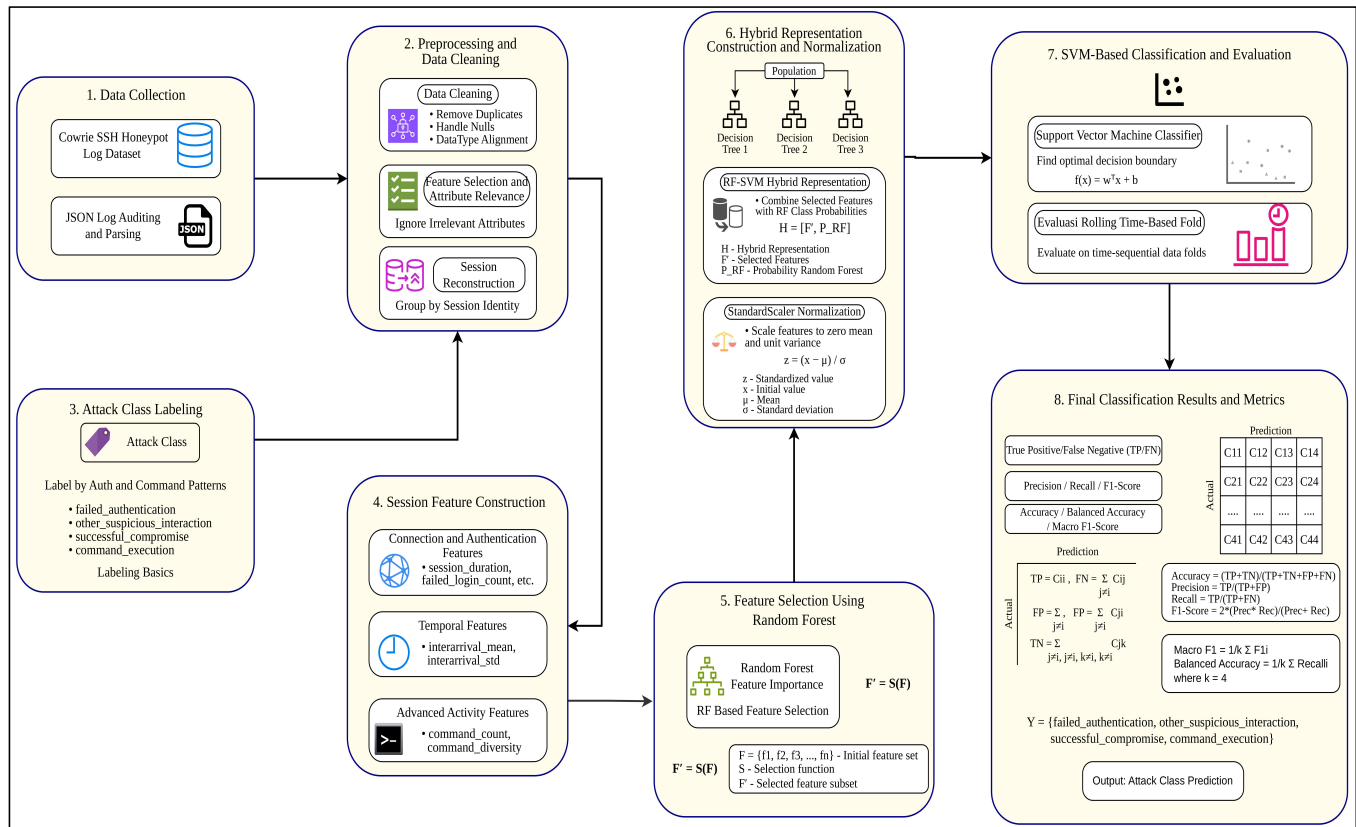


Fig. 1. Workflow of the hybrid random forest-support vector machine-based attack classification method.

hybrid approach. Random Forest-Support Vector Machine approach, and evaluation through a rolling time-based fold scheme. Each stage is structured to transform attacker activities that were initially recorded at the event level into a more compact, organized, and informative session representation. This representation supports more reliable model evaluation and provides a clearer interpretation of each methodological stage. The overall workflow is presented in Fig. 1.

Data was obtained from a Cowrie SSH HoneyPot deployed on a CloudHost VPS to record activities targeting SSH services. The honeypot was used because it provides a controlled environment for attracting attacker interactions and capturing authentication activities, network communication, and follow-up actions during sessions. This approach aligns with the concept of cyber deception, in which honeypots are used to observe attack behavior more specifically [3], [4]. Cowrie logs contain important information, including timestamps, event IDs, source IP addresses, ports, session identifiers, authentication status, connection duration, and commands submitted during interactions. The audit results showed that the dataset consisted of 38 log files, 213,494 valid records, 29,099 unique sessions, and 1,618 unique source IP addresses. All valid records were used to construct the session-based dataset.

The cleaning stage was performed by reading all logs in

JSON format, removing duplicates, handling missing values, and aligning data types. Attributes such as timestamp, eventid, script, session, username, password, and input were retained because they directly support the construction of session features. Preprocessing, feature selection, and normalization can affect IDS performance, especially when the data are high-dimensional and exhibit diverse characteristics [21]. Realistic and structured cybersecurity datasets are important for supporting data-driven analysis, while temporal characteristics are also essential for interpreting attack patterns [9], [10]. Through this stage, raw data can be transformed into a more consistent representation for session reconstruction and classification. After cleaning, sessions were reconstructed using Cowrie's native session identifiers by chronologically ordering the associated events and calculating duration from the first to the last recorded timestamp. Class labels were assigned based on authentication patterns and command activity within each session. Sessions with successful login followed by command activity were labeled as command execution. Sessions with successful login but without command activity were labeled as successful compromise. Sessions without successful login but containing failed login attempts were labeled as failed authentication.

Sessions that did not match these patterns were labeled as other suspicious interaction. In the initial classification stage, command execution and successful compromise were combined into successful interaction. The classification output was then refined into the four final classes. This labeling strategy was applied to help the model distinguish sessions that only contain authentication attempts, sessions that indicate successful access, and sessions that have reached the command execution stage.

Session features were constructed to represent four main aspects of attack behavior: connection, authentication, temporal activity, and post-connection activity. The connection aspect was represented by session duration, event count, and source repetition. The authentication aspect included the failed login count, successful login count, login success rate, username diversity, and password diversity. The interarrival mean and standard deviation represented the temporal aspect. At the same time, post-connection activity was described through command count, command diversity, client version seen, client kex seen, and direct tcpip request count. This combination of features enabled the model to capture not only the final session status but also connection intensity, credential variation, temporal rhythm, and behavioral patterns that emerged after the connection was established.

Feature selection was performed to remove less informative attributes and retain features with the strongest contribution to distinguishing attack classes. Cybersecurity data often contains diverse features, some of which may be irrelevant and could increase model complexity if used without filtering. Feature selection can reduce data dimensionality and preserve more informative attributes in high-dimensional data [12]. It also supports the identification of important attributes in machine-learning-based IDS [22]. Random Forest combined with feature selection can support intrusion detection, while feature stability remains an important aspect of machine learning-based IDS [23], [24]. Random Forest selected the five highest-ranked features from the training data for each rolling fold to prevent test data leakage. The feature selection process is expressed in Equation (1).

$$F^* = \Phi_m(F) \quad (1)$$

In Equation (1), F represents the ten candidate features, Φ_m represents the Random Forest-based ranking function applied to each training fold, and F^* represents the five-feature subset retained for hybrid modeling. The model used in this study is the Hybrid Random Forest-Support Vector Machine. Within each outer training fold, Random Forest ranked ten candidate behavioral features, retained the five highest-ranked features, and generated time-aware out-of-fold class-probability estimates. The selected features and RF probabilities were standardized and passed to the SVM as the Stage-1 classifier. Random Forest-based approaches have been applied in attack and threat detection [13], [17]. Hybrid feature selection and ensemble learning approaches can support the development of stronger IDS, especially in multiclass scenarios and imbalanced data conditions [25]. Random Forest predictions are generated through majority voting from multiple decision trees, as expressed in Equation (2).

$$\hat{y} = \text{mode}(h_1(x), h_2(x), \dots, h_B(x)) \quad (2)$$

In Equation (2), $h_B(x)$ represents the prediction of the B -th decision tree, while $\hat{y}$ denotes the class obtained from the voting process. After the selected features were obtained, the Random Forest's probability output was combined with them to form a hybrid representation. This representation was then scaled using StandardScaler before being used as input for SVM.

A Support Vector Machine was used as the Stage-1 classifier, with the decision function given by Equation (3).

$$f(x) = w^T x + b \quad (3)$$

In Equation (3), w represents the weight, x represents the feature vector, and b represents the bias. Random Forest was configured with 200 decision trees, `max_depth` of 8, `min_samples_split` of 6, `min_samples_leaf` of 3, and `class_weight` set to `balanced_subsample`. SVM used an RBF kernel, `C=0.8`, `gamma` set to `scale`, `class_weight` set to `balanced`, and `probability=True`. The use of `class_weight` in both models was intended to reduce the influence of dominant classes during training. The Stage-1 outputs comprised three classes: `failed_authentication`, `other_suspicious_interaction`, and `successful_interaction`; the latter was refined into `command_execution` or `successful_compromise` based on recorded command activity.

Model evaluation was conducted using a rolling time-based fold scheme. The configuration consisted of 14 days of training data, 3 days of testing data, and a 3-day shift for each fold. This scheme produced eight valid folds. Time-based data splitting was selected because attack activity in a honeypot environment is dynamic, requiring the model to be tested across moving time windows rather than only with random data splitting. The characteristics of attacks may change over time and across network contexts, making time-based evaluation more appropriate for cybersecurity data. Discussions on intrusion detection and temporal approaches in network anomaly analysis also support the importance of time-based evaluation for cybersecurity data [19], [10].

Model performance was evaluated using several metrics, including accuracy, precision, recall, F1-score, balanced accuracy, macro F1-score, and ROC-AUC. Since the dataset has an imbalanced class distribution, balanced accuracy and macro F1-score were used as important complementary metrics in addition to accuracy. The basic formulas for the evaluation metrics are presented in Equations (4)-(7).

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

$$\text{Precision} = \frac{TP}{(TP + FP)} \quad (5)$$

$$\text{Recall} = \frac{TP}{(TP + FN)} \quad (6)$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (7)$$

In Equations (4) to (7), TP, TN, FP, and FN represent true positive, true negative, false positive, and false negative, respectively. The evaluation results were then analyzed using the performance summary for each fold, the aggregate confusion matrix, class distribution, and the stability of the selected features. This analysis was conducted not only to assess the overall model performance, but also to examine the model's consistency in recognizing each attack class.

IV. RESULTS AND DISCUSSION

The dataset used in this article was obtained from Cowrie SSH Honeypot logs that had undergone auditing, data cleaning, and session reconstruction. This process produced 38 log files containing 213,494 valid records, 29,099 unique sessions, and 1,618 unique source IP addresses. All readable records were valid, allowing the entire dataset to be used as the basis for constructing session-based data. The number of generated sessions indicates that the dataset is not only large in volume but also reflects diverse attacker interactions with SSH services. The large number of source IP addresses further indicates that the recorded activities originated from multiple connection points, making the dataset sufficiently representative to support a comprehensive analysis of session-based attack behavior. A summary of the dataset components is presented in Table 1.

Table 1.
Dataset Summary

Components	Value
Total file logs	38
Total record valid	213.494
Total unique sessions	29.099
Total unique source IP addresses	1.618

Table 1 shows that the dataset has sufficient volume and diversity to support session-based attack classification. The presence of valid records, unique sessions, and varied source IP addresses indicates that the reconstructed dataset can capture different patterns of attacker interactions with SSH services.

The final class distribution shows a clear imbalance, as summarized in Table 2. The dataset is dominated by command execution, while failed authentication, other suspicious interactions, and successful compromise appear in much smaller proportions. This imbalance indicates that accuracy alone is insufficient to capture overall model performance, so balanced accuracy and macro F1-score were used to provide a fairer assessment across all classes.

Table 2 indicates that command execution accounts for the largest portion of the final class distribution, whereas the other three classes contain considerably fewer samples. This imbalance shows that minority classes need to be evaluated carefully. Therefore, model performance should be judged not only by accuracy but also by metrics that assess each class more proportionally. Such evaluation helps minimize bias toward the

dominant class and supports a more balanced interpretation of model performance, particularly under imbalanced data conditions.

Table 2.
Final class distribution

Attack class	Number of sessions	Percentage
Command execution	23,427	80.51
Failed authentication	3,222	11.07
Other suspicious interaction	1,260	4.33
Successful compromise	1,190	4.09

Figure 2 illustrates the dominance of command execution compared with the other attack classes. This visual pattern indicates that most sessions in the honeypot environment progressed to command execution, while fewer sessions remained at the successful compromise stage without visible command activity. This condition makes the final four-class classification more challenging because the model must distinguish sessions with similar initial access patterns but different follow-up behaviors.

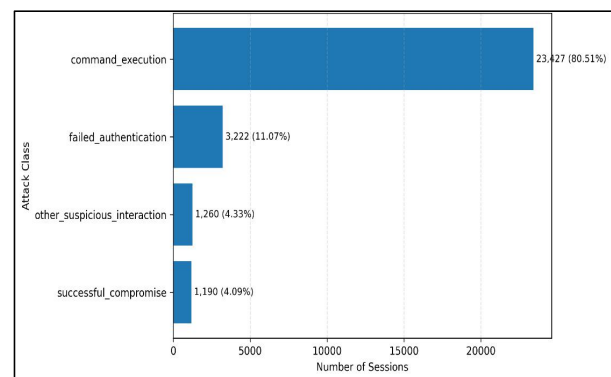


Fig. 2. Final class distribution of session-based SSH attacks.

Five features were consistently selected across all folds, namely password diversity, interarrival std, username diversity, interarrival mean, and session duration. Although the initial configuration allowed selecting up to 8 top features, the final implementation used only 5 effective candidate features after several attributes were removed from the training process. Table 3 shows that four features, namely event_count, username_diversity, interarrival_std, and password_diversity, were consistently selected across all 8 folds, while source_repetition and interarrival_mean were selected in 6 and 2 folds, respectively.

This selection pattern indicates that the model relied on stable indicators rather than on features that were useful only in certain periods. Therefore, the four consistently selected features can be considered representative attributes for distinguishing session-based attack behavior. The repeated selection of these features also suggests that session activity volume, credential variation, and temporal variability provide reliable signals for separating attack classes. This strengthens the interpretation that the model captures meaningful behavioral patterns rather than fold-specific attributes. This feature-selection pattern is further illustrated in Fig. 3.

Table 3.
Feature Selection Frequency Across Folds

Features	Number of Folds
event_count	8
username_diversity	8
interarrival_std	8
password_diversity	8
source_repetition	6
interarrival_mean	2

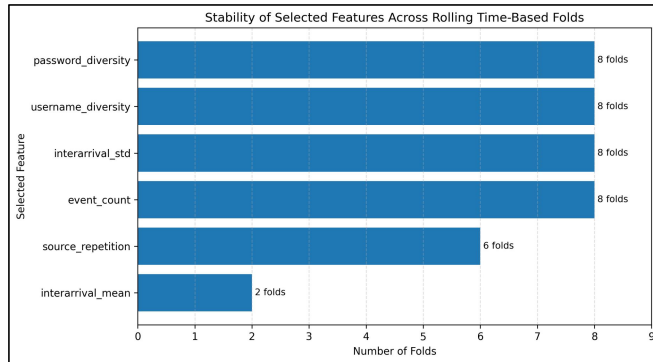


Fig. 3. Feature selection frequency across eight rolling folds.

Figure 3 confirms that event count, username diversity, interarrival_std, and password diversity were selected in all eight folds, while source repetition and interarrival mean appeared in six and two folds, respectively. These results identify session activity, credential variation, and temporal variability as the most stable behavioral signals.

Model evaluation was conducted using a rolling time-based fold scheme, with 14 days of data used for training, the next 3 days for testing, and a 3-day shift for each fold. This scheme produced eight valid folds and was used because honeypot attack activity can change over time. By using time-based splitting, the model was tested across moving periods, making the evaluation closer to operational conditions than random splitting. This evaluation pattern is important for cybersecurity data because attack characteristics may change over time, as attack sources and the intensity of activity entering the system evolve.

In stage 1, the model distinguished three main groups: failed authentication, other suspicious interactions, and successful interactions. The revised evaluation results showed an accuracy of $99.65\% \pm 0.38\%$, a balanced accuracy of $98.89\% \pm 1.39\%$, a macro F1-score of $99.26\% \pm 0.99\%$, and a macro ROC-AUC of $99.89\% \pm 0.29\%$. These results indicate that the model effectively separated the three initial groups. The role of Random Forest in feature selection and time-aware class-probability generation helped provide a compact hybrid representation for the SVM Stage-1 classifier.

After rule-based refinement into four final classes, the model maintained an accuracy of $99.65\% \pm 0.38\%$. However, balanced accuracy decreased to $97.02\% \pm 5.77\%$, while macro F1-score reached $97.84\% \pm 4.37\%$. This decrease does not indicate model failure but shows that the final classification task was more difficult. In Stage-1, command execution and successful compromise were still grouped in the successful interaction group. After refinement, these two classes were

separated into different categories. This separation was more challenging because both classes began with successful authentication but differed in whether command execution occurred after login. A summary of the model performance is presented in Table 4.

Table 4.
Model Performance Summary

Classification stage	Accuracy (%)	Balanced Accuracy (%)	Macro F1-score (%)	ROC-AUC Macro (%)
Stage-1	99.65 ± 0.38	98.89 ± 1.39	99.26 ± 0.99	99.89 ± 0.29
Final 4 classes	99.65 ± 0.38	97.02 ± 5.77	97.84 ± 4.37	—

Table 4 shows that Stage-1 achieved stronger balanced accuracy and macro F1-score than the final four-class classification. Although accuracy remained the same in both stages, the decrease in balanced accuracy and macro F1-score indicates that separating successful interaction into command execution and successful compromise increased the classification difficulty. This comparison confirms that the final classification requires the model to distinguish more specific behavioral differences between classes. To provide a clearer comparison between Stage-1 and the final four-class classification, the performance results are visualized in Fig. 4.

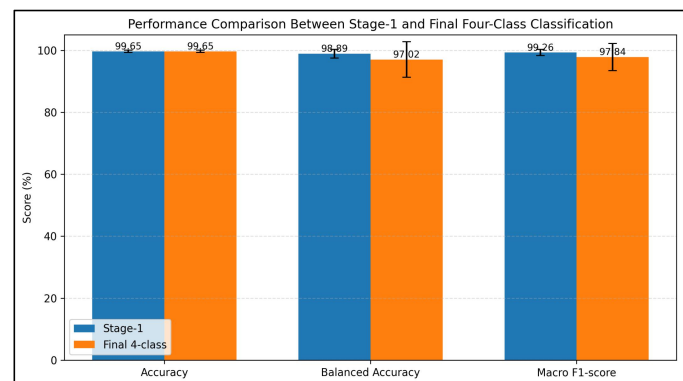


Fig. 4. Performance comparison between Stage-1 and final four-class classification.

The difference between Stage-1 and the final four-class output shows that accuracy alone is not sufficient for evaluating model performance. Although accuracy remained high, the dominance of the majority class may obscure performance in minority classes. Therefore, balanced accuracy and macro F1-score provide a more proportional evaluation because they consider performance across all classes. The result indicates that the model maintained strong performance on the imbalanced data but faced greater difficulty when separating classes with similar initial patterns. This condition confirms that honeypot-based model evaluation should be interpreted using multiple metrics. The final classification performance across each fold is presented in Fig. 5.

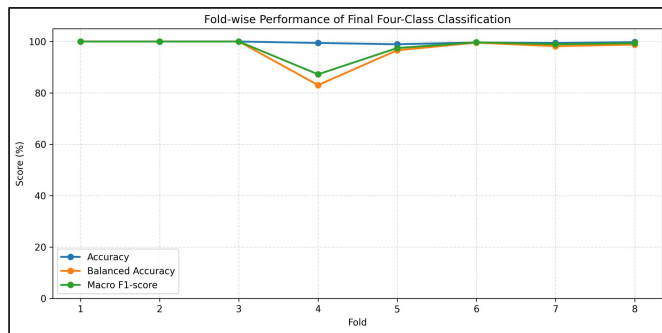


Fig. 5. Final classification performance on each fold.

Figure 5 shows that model performance varied across folds because the attack patterns recorded in the honeypot did not appear with the same class composition in each testing period. The number and distribution of classes varied from one-fold to another, leading to differences in classification difficulty. The largest decline occurred in Fold 4, where accuracy remained at 98.93%, while balanced accuracy and macro F1-score decreased to 83.05% and 87.24%, respectively. Performance recovered in the subsequent folds, indicating that the decline was limited to a specific testing period rather than representing persistent model degradation. These findings reinforce the importance of time-based evaluation, as the model was tested using the chronological order of attack data rather than random data splits.

The aggregate confusion matrix is presented in Fig. 6 further to examine the prediction pattern of the final model. The matrix shows that most predictions were located on the main diagonal, indicating that the model recognized most attack classes well. The other suspicious interaction class showed strong performance, with 1,033 correct predictions and only 2 sessions misclassified as successful compromise. The command execution class also showed strong performance, with 19,689 correct predictions and only 2 sessions misclassified as failed authentication. This pattern indicates that sessions containing command activity have clearer behavioral characteristics than sessions containing only authentication activity.

Misclassification mainly occurred between successful compromise and failed authentication. In the revised evaluation, 14 successful compromise sessions were misclassified as failed authentication, while 867 were correctly classified. The misclassified sessions were shorter, with a mean duration of 5.53 seconds and a median of 1.78 seconds, compared with 7.88 seconds and 7.56 seconds, respectively, for correctly classified sessions. This difference supports the interpretation that shorter sessions with limited observable activity were more difficult to distinguish from failed authentication. Misclassification mainly occurred between successful compromise and failed authentication. In the revised evaluation, 14 successful compromise sessions were misclassified as failed authentication, while 867 were correctly classified. The misclassified sessions were shorter, with a mean duration of

5.53 seconds and a median of 1.78 seconds, compared with 7.88 seconds and 7.56 seconds, respectively, for correctly classified sessions. This difference supports the interpretation that shorter sessions with limited observable activity were more difficult to distinguish from failed authentication.

From the modeling perspective, these results show that Random Forest and SVM play complementary roles in the classification workflow. Random Forest ranked behavioral features and generated time-aware class-probability estimates, while SVM used the combined representation for Stage-1 classification. The consistent selection of event count, username diversity, interarrival_std, and password diversity indicates that session activity, credential variation, and temporal variability were the most stable signals.

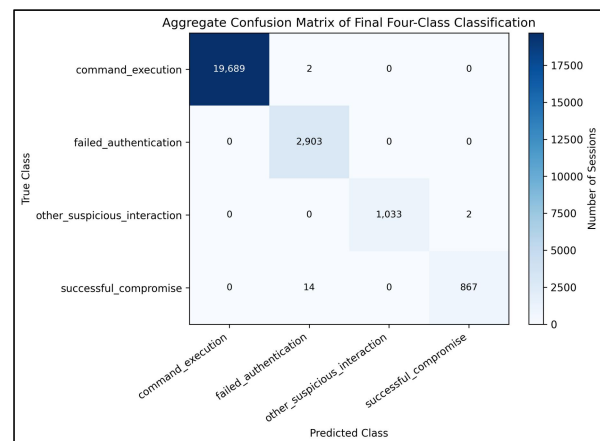


Fig. 6. Aggregate confusion matrix of the Hybrid RF-SVM model.

The findings of this study are consistent with previous intrusion-detection research indicating that feature selection and data preprocessing strategies can support classification under complex and imbalanced conditions [20], [21]. However, the present study differs in that it focuses on session-level behavioral data derived from Cowrie SSH Honeypot logs and combines RF-based feature selection, time-aware out-of-fold probability generation, and RBF-SVM classification with rolling time-based evaluation. Direct numerical comparison should be interpreted cautiously because the datasets, class definitions, and evaluation protocols differ.

From an application perspective, the Hybrid RF-SVM framework can support honeypot log triage by grouping sessions based on recorded activity levels. Failed authentication can indicate credential attempts; successful compromise can indicate access without further activity; and command execution can indicate higher-risk sessions involving command execution. Although the results show strong performance, the dataset was obtained from a single Cowrie SSH Honeypot environment, and the final labels were based on heuristic session-level rules. Therefore, future evaluation can be expanded by using multi-honeypot datasets, expert-labeled validation, unseen-IP scenarios, longer observation periods, scalability testing, and integration with IDS or SIEM platforms to assess model generalization more thoroughly.

V. CONCLUSION

This study demonstrates that a Hybrid RF–SVM framework can transform Cowrie SSH Honeypot event logs into interpretable session-level attack classifications. The main methodological contribution lies in integrating RF-based feature selection and time-aware out-of-fold probability generation with SVM Stage-1 classification and rule-based refinement. Across the rolling time-based evaluation folds, event count, username diversity, interarrival_std, and password diversity remained the most stable indicators, showing that session activity volume, credential variation, and temporal variability are important for distinguishing SSH attack behavior. The analysis also indicates that successful compromise sessions with shorter durations and limited observable activity are more difficult to distinguish from failed authentication.

In practice, the framework can support honeypot log triage by grouping sessions based on authentication outcomes and command activity, thereby helping security analysts prioritize higher-risk interactions. However, the findings are limited using data from a single Cowrie environment, heuristic session-level labels, and aggregated behavioral features that do not fully preserve the order of events. Future studies should evaluate multi-honeypot datasets, expert-validated labels, unseen-IP scenarios, longer observation periods, scalability, and integration with IDS or SIEM platforms. Sequential or temporal models should also be investigated to capture event-order information and changes in attack behavior over time.

ACKNOWLEDGMENT

The authors express their appreciation to the Faculty of Engineering at the University of Muhammadiyah Makassar for the support, facilities, and academic environment provided during the preparation of this article.

REFERENCES

- [1] R. Kaur, D. Gabrijelčič, and T. Klobočar, “Artificial intelligence for cybersecurity: Literature review and future research directions,” *Inf. Fusion*, vol. 97, Art. no. 101804, Apr. 2023, doi: 10.1016/j.inffus.2023.101804.
- [2] O. Alnasser, J. Al Muhtadi, K. Saleem, and S. Shrestha, “Signature and anomaly based intrusion detection system for secure IoTs and V2G communication,” *Alexandria Eng. J.*, vol. 125, pp. 424–440, Apr. 2025.
- [3] M. Alshehri, “Dynamic cyber deception using AI-driven adaptive honeypot networks and context-aware behavioural analysis for proactive threat neutralization,” *Ain Shams Eng. J.*, vol. 17, no. 4, Art. no. 104061, Mar. 2026, doi: 10.1016/j.asej.2026.104061.
- [4] P. Beltrán-López, M. Gil Pérez, E. Vasilomanolakis, and P. Nespoli, “Reactive cyber deception: Stealth-based adaptive redirection to on-demand honeypots with AI-driven data generation,” *Comput. Networks*, vol. 281, Art. no. 112203, Mar. 2026, doi: 10.1016/j.comnet.2026.112203.
- [5] A. Javadpour, F. Ja, T. Taleb, M. Shojafar, and C. Benzaïd, “A comprehensive survey on cyber deception techniques to improve honeypot performance,” *Comput. Secur.*, vol. 140, Art. no. 103792, Mar. 2024, doi: 10.1016/j.cose.2024.103792.
- [6] N. Ambrosio, C. Lista, G. Perrone, and S. Pietro Romano, “SMASH: An SDN-MTD framework for efficient honeypot deployment and insider threat mitigation,” *Comput. Networks*, vol. 269, Art. no. 111327, Jun. 2025, doi: 10.1016/j.comnet.2025.111327.
- [7] S. Dadkhah *et al.*, “CIC-YNU-IoTMal: A comprehensive multilayer dataset for static and dynamic analysis of IoT malware behavior,” *Inf. Syst.*, vol. 140, Art. no. 102722, Apr. 2026, doi: 10.1016/j.is.2026.102722.
- [8] V. Boiko and O. Niiakyi, “A 4-Month Dataset of Interactive SSH Honeypot Logs and Command Payloads,” *Data Br.*, vol. 66, Art. no. 112860, May 2026, doi: 10.1016/j.dib.2026.112860.
- [9] W. Borges, G. Ventorim, R. Vieira, I. Drago, and S. Villça, “Building realistic cybersecurity datasets through a edge cloud testbed with cost-aware monitoring,” *Comput. Networks*, vol. 282, Art. no. 112303, Apr. 2026, doi: 10.1016/j.comnet.2026.112303.
- [10] M. Ozdem, “A novel approach for real-time anomaly detection in dynamic computer networks using temporal graph networks and explainable artificial intelligence,” *Alexandria Eng. J.*, vol. 132, pp. 369–382, Nov. 2025, doi: 10.1016/j.aej.2025.11.001.
- [11] M. A. Shyaa, N. Farizah, Z. Zainol, R. Abdullah, and L. Alzubaidi, “Evolving cybersecurity frontiers: A comprehensive survey on concept drift and feature dynamics aware machine and deep learning in intrusion detection systems,” *Eng. Appl. Artif. Intell.*, vol. 137, Art. no. 109143, Aug. 2024, doi: 10.1016/j.engappai.2024.109143.
- [12] N. Garcia-pedrajas and G. Cerruela-garcía, “MABUSE: A margin optimization based feature subset selection algorithm using boosting principles,” *Knowledge-Based Syst.*, vol. 253, Art. no. 109529, 2022, doi: 10.1016/j.knsys.2022.109529.
- [13] J. Bhayo, S. Attique, S. Hameed, A. Ahmed, J. Nasir, and D. Draheim, “Towards a machine learning-based framework for DDOS attack detection in software-defined IoT (SD-IoT) networks,” *Eng. Appl. Artif. Intell.*, vol. 123, Art. no. 106432, May 2023, doi: 10.1016/j.engappai.2023.106432.
- [14] I. Akbar and M. Faisal, “Perbandingan Analisis Sentimen PLN Mobile: Machine Learning vs. Deep Learning,” *JOINTECS (Journal Inf. Technol. Comput. Sci.)*, vol. 8, no. 1, pp. 1–10, 2024.
- [15] Sarina *et al.*, “Klasifikasi Penyakit Tanaman Nilam Berdasarkan Citra Daun Menggunakan GLCM dan SVM,” *J. Inform. Prog.*, vol. 17, no. 2, pp. 12–22, 2025, doi: 10.56708/progres.v17i2.469.
- [16] M. Faisal *et al.*, “Utilizing Machine Learning-Based Decision- Making to Align Higher Education Curriculum with Industry Requirements,” *Int. J. Mod. Educ. Comput. Sci.*, vol. 17, no. 4, pp. 1–25, 2025, doi: 10.5815/ijmecs.2025.04.01.
- [17] Z. Noor, S. Hina, F. Hayat, and G. A. Shah, “Internet of Things An intelligent context-aware threat detection and response model for smart cyber-physical systems,” *Internet of Things*, vol. 23, Art. no. 100843, Jun. 2023, doi: 10.1016/j.iot.2023.100843.
- [18] A. Singh, P. K. Chouhan, and G. S. Aujla, “SecureFlow: Knowledge and data-driven ensemble for intrusion detection and dynamic rule configuration in software-defined IoT environment,” *Ad Hoc Networks*, vol. 156, Art. no. 103404, 2024, doi: 10.1016/j.adhoc.2024.103404.
- [19] P. Sarantos, J. Violos, and A. Leivadadas, “Enabling semi-supervised learning in intrusion detection systems,” *J. Parallel Distrib. Comput.*, vol. 196, Art. no. 105010, Nov. 2024, doi: 10.1016/j.jpdc.2024.105010.
- [20] T.-T.-H. Le, Y. Shin, M. Kim, and H. Kim, “Towards unbalanced multiclass intrusion detection with hybrid sampling methods and ensemble classification,” *Appl. Soft Comput.*, vol. 157, Art. no. 111517, 2024, doi: 10.1016/j.asoc.2024.111517.
- [21] M. A. Umar, Z. Chen, K. Shuaib, and Y. Liu, “Effects of feature selection and normalization on network intrusion detection,” *Data Sci. Manag.*, vol. 8, no. 1, pp. 23–39, 2025, doi: 10.1016/j.dsm.2024.08.001.
- [22] N. O. Aljehane, H. A. Mengash, S. B. H. Hassine, F. A. Alotaibi, A. S. Salama, and S. Abdelbagi, “Optimizing intrusion detection using intelligent feature selection with machine learning model,” *Alexandria Eng. J.*, vol. 91, pp. 39–49, Feb. 2024, doi: 10.1016/j.aej.2024.01.073.
- [23] A. A. Wardana, P. Sukarno, S. Basuki, and S. Budhi, “Federated Random Forest with Feature Selection for Collaborative Intrusion Detection in Internet of Things,” *Procedia Comput. Sci.*, vol. 246, pp. 20–29, 2024, doi: 10.1016/j.procs.2024.09.193.
- [24] T. Sowmya and M. A. E. A., “A novel stable feature selection algorithm for machine learning based intrusion detection system,” *Procedia Comput. Sci.*, vol. 252, pp. 738–747, Jan. 2025, doi: 10.1016/j.procs.2025.01.034.
- [25] A. Wakili and S. Bakkali, “A resilient IoT intrusion detection system using hybrid feature selection and explainable ensemble learning,” *Results Eng.*, vol. 28, Art. no. 107392, Oct. 2025, doi: 10.1016/j.rineng.2025.107392.