Matrik: Jurnal Manajemen, Teknik Informatika, dan Rekayasa Komputer
Vol. 25, No. 1, November 2025, pp. 145∼160
ISSN: 2476-9843, accredited by Kemenristekdikti, Decree No: 200/M/KPT/2020
DOI: 10.30812/matrik.v25i1.5817

Improving Detection Accuracy of Brute-Force Attacks on MariaDB Using Standard Isolation Forest: A Comparative Analysis with Rotated Variant

Hartono[1], Khusnul Khotimah[1], Rokin Maharjan[2]
[1]Information Systems and Technology, Muhammadiyah University of Kotabumi, Indonesia
[2]Department of Computer Science, Baylor University, United States

Article Info

Article history:

Received October 09, 2025
Revised November 04, 2025
Accepted November 21, 2025

Keywords:

Anomaly Detection;
Brute Force;
Isolation Forest;
Log Analysis;
Rotated Isolation Forest.

ABSTRACT

Brute-force attacks remain among the most prevalent and persistent cybersecurity threats to database systems, causing unauthorized access, data leakage, and service disruptions. Conventional threshold-based detection methods often struggle to adapt to evolving and dynamic attack patterns, necessitating more robust anomaly detection approaches. This study aims to develop, evaluate, and compare two unsupervised machine learning algorithms Standard Isolation Forest (IF) and Rotated Isolation Forest (RIF) for detecting brute-force attacks targeting databases such as MariaDB. A large-scale raw access log dataset containing millions of entries was pre-processed through data cleaning, normalization, and feature extraction. Behavioural features were engineered for IP-path pairs, including login-attempt frequency, request intervals, and rapid-attempt ratios. The dataset consisted of 1,831,989 benign and 5,126,052 brute-force entries. The Standard IF model was trained using benign data (n_estimators = 175, contamination = 0.1, max_samples = 'auto') and evaluated on mixed data, achieving Recall 99.94%, Precision 99.29%, F1-Score 99.61%, AUC 0.9495, and Accuracy 99.28%, with TP = 5,123,224 and FN = 2,828. The RIF model, using Gaussian Random Projection (n_components = 5), yielded slightly lower metrics: Recall 99.44%, F1-Score 99.36%, and Accuracy 98.81%. The findings indicate that Standard Isolation Forest provides higher detection accuracy and reliability in identifying brute-force anomalies within large-scale log data. Despite the theoretical advantage of feature rotation in handling complex anomalies, the Standard IF demonstrates superior practical performance and efficiency. Overall, the study confirms the method's strong potential for integration into automated and real-time cybersecurity monitoring systems.

Corresponding Author:

Hartono, +6281997112904
Faculty of Engineering and Computer Science, Study Program of Information Systems and Technology,
Universitas Muhammadiyah Kotabumi, Lampung, Indonesia,
Email: hartono@umko.ac.id

How to Cite:
Hartono, K. Khotimah, and R. Maharjan, "Improving Detection Accuracy of Brute-Force Attacks on MariaDB Using Standard Isolation Forest: A Comparative Analysis with Rotated Variant", *MATRIK: Jurnal Manajemen, Teknik Informatika, dan Rekayasa Komputer*, Vol. 25, No. 1, pp. 145-160, November, 2025.

Journal homepage: *https://journal.universitasbumigora.ac.id/index.php/matrik*

1. INTRODUCTION

Brute force attacks are a relatively simple yet highly effective and destructive method. This attack systematically attempts combinations of usernames and passwords or other authentication keys until the correct pair is found [1–3]. According to The Shadowserver Foundation, large-scale brute force attacks have utilized approximately 2.8 million distinct IP addresses per day in attempts to guess credentials [3]. Other studies report similar scales, with over 1.3 billion unique username-password pairs found in breach compilations [4]. The Canva breach shows how brute-force-based credential cracking and credential stuffing enabled mass account compromise, underscoring the need for strong authentication defenses [5]. These attacks frequently occur on a large scale, either originating from a single source or distributed across numerous IP addresses, and may transpire over a short period or remain concealed with low frequency [6–8]. Honeypot studies have identified collusion among several seemingly unrelated IP addresses, indicating the presence of coordinated attacks [8]. Although brute-force detection has been an area of research and security solution development, traditional methods often rely on signature- or threshold-based rules. These methods are effective against known attack patterns or those that clearly deviate. Still, they tend to struggle to detect more sophisticated attacks that may employ low-and-slow techniques or pattern variations to evade static detection. These limitations create gaps in the ability to detect brute force attacks. Brute force attacks are highly dangerous because they can be executed automatically at high speed using specialised tools or botnets. Moreover, the sheer volume and persistence of these attacks put additional strain on server resources, potentially degrading overall system performance [9].

Recent studies highlight that brute-force attacks remain a persistent threat in production environments, with occurrences ranging between 6,470 and 22,715 attacks per server per day [10]. These attacks often target network service accounts and may originate both externally and internally [10–12]. Early detection enables system administrators to promptly implement mitigation actions, such as blocking attacker IP addresses, locking suspicious accounts, or strengthening security policies, thereby minimizing potential damage. As mentioned [12], early detection is essential to enable rapid mitigation actions, to reduce system compromise. In response, researchers have increasingly adopted unsupervised learning approaches for intrusion detection, which learn normal behavior from unlabeled data and identify deviations as anomalies [13]. IF algorithm operates by isolating anomalous observations within random decision trees, where anomalies tend to be more easily isolated compared to normal data [14]. The superiority of IF lies in its efficiency and its capability to handle high-dimensional data [13–15].

The objective of this study is to evaluate the effectiveness of IF algorithm in detecting brute-force attacks on MariaDB by measuring its detection accuracy, false-positive rate, and overall performance. Additionally, this study aims to compare these results with those produced by the Rotated IF variant to determine whether the rotational transformation provides measurable improvements. The main contributions of this study are: (1) developing brute-force attack detection models based on IF and Rotated IF optimized for MariaDB log data, (2) conducting a comparative evaluation using standard anomaly detection metrics, and (3) providing practical recommendations for deploying automated anomaly detection in production database systems. Brute force attacks has the presence of deviations in access data, which deviates from this norm is referred to as anomalies or outliers [16]. Anomaly detection holds significant relevance in identifying new attacks or variants that are difficult to recognize, such as zero-day attacks and evasive attacks, which are generally not detected by signature-based detection systems due to their limitations in recognizing only known attack patterns [17]. IF was chosen as the model due to the high volume of access log data, which cannot be entirely labelled manually [18].**RIF was first introduced in a paper published on January 29, 2025, by Vahideh Monemizadeh and Kourosh Kiani. The algorithm was developed in response to the remaining issues in EIF, particularly ghost inter-clusters, and proposes a solution by applying random rotations to the dataset before the isolation process [19, 20]**.

Several previous studies have used IF to detect cyberattacks. Study [14] employed the IF method for detecting various attacks in Industrial Control Systems (ICS) and achieved a recall of 80%. Furthermore, study [21], applied the IF algorithm for anomaly detection, obtaining an accuracy of 98.9%, precision of 97.9%, and recall of 98.5%. In study [22] IF was used to detect anomalies in web traffic, yielding an accuracy of 93%, precision of 95%, and recall of 90%. In the Bot-IoT case [22], a combination of IF and Random Forest achieved an accuracy of 99.98%. Study [13] noted that detection using IF is highly adaptive and has the potential for high accuracy, but requires careful threshold tuning, especially when dealing with imbalanced data. Study [15] employed Deep IF and reported an AUC-ROC of 0.925. IF and its variants consistently demonstrate high detection accuracy. The Hybrid Density-Based IF achieved 98.9% accuracy, 97.9% precision, 98.5% recall, and 98.6% F1-score on the NSL-KDD, CICIDS2017, and UNSW-NB15 datasets [22]. For web traffic anomaly detection, IF recorded an accuracy of 93%, precision of 95%, recall of 90%, and F1-score of 92% [23]. In ICS, the Dual IF model was capable of detecting attacks missed by other methods, achieving a minimum recall of 80% and improving the F1-score by up to 7% [14].

IF randomly chooses a split value for the selected feature, which lies between the minimum and maximum values of that feature in the chosen data subset. Data partitioning is performed recursively until each data point is isolated or the maximum tree depth is reached [15]. Normal data requires more partitions to be isolated because it tends to reside in dense, similar clusters. In contrast, anomalies, which are located far from normal data and possess unique characteristics, typically require fewer partitions to be isolated [24]. Research specifically addressing brute force attacks on MariaDB remains limited; however, several relevant findings have been reported. Brute force attacks targeted a web server using MariaDB [25]. Studies also show that other techniques, including compression and encryption, can expose sensitive database information [26]. Most brute force attacks on MariaDB do not target the server directly. Instead, detection relies on reviewing application or web server access logs, which track login attempts and help reveal attack patterns. MariaDB produces more diverse and detailed log entries than MySQL or PostgreSQL, including authentication attempts and query-level events across multiple log types. The variability in log formats and verbosity levels complicates automated parsing and anomaly detection. Moreover, its pluggable authentication mechanisms produce inconsistent login patterns, making the detection of brute-force attacks more challenging.

IF has demonstrated strong computational efficiency, particularly for high-dimensional, large-volume datasets, as it does not require computing distances between all data points. Additionally, the algorithm is capable of handling irrelevant features and is less sensitive to outliers [24]. IF often produces empty regions that are mistakenly identified as normal clusters, resulting in biased anomaly scores and less accurate predictions, particularly in high-dimensional data or those with complex distributions [19]. Rotated IF (RIF) addresses empty clusters by allowing data splits beyond axis-aligned orientations, improving class separability and anomaly detection. This can help identify attack patterns that standard IF misses, especially in high-dimensional data. This study will compare the effectiveness of IF and RIF in detecting brute-force attacks.

Unlike prior studies that primarily applied Isolation Forest on general web traffic, IoT, or network intrusion datasets, this study introduces a domain-specific adaptation of IF and RIF for large-scale MariaDB access logs. The dataset incorporates domain-level features such as IP-path behavioral metrics, attempt frequency ratios, and request interval dynamics, which have not been previously explored in brute-force detection. Moreover, the research implements a comparative evaluation under identical configurations and dynamic contamination thresholds to assess model sensitivity to real-world log imbalance. This integrative approach provides new empirical evidence on the performance boundaries of Standard and Rotated Isolation Forests in detecting brute-force anomalies within database environments.

2. RESEARCH METHOD

This study adopts an unsupervised learning approach using the IF algorithm. IF was selected due to its extensive application in detecting anomalies across various cybersecurity domains, such as network intrusion detection, malware identification, web traffic anomaly detection, email security, and critical infrastructure protection [21, 23, 27]. The selection of this methodology is based on the need to effectively detect anomalies in log data that are large in size, complex, and do not always have explicit labels [28–30]. With this approach, the research is expected to yield a robust, systematic, and reproducible detection system. The method's implementation flow is shown in Figure 1.

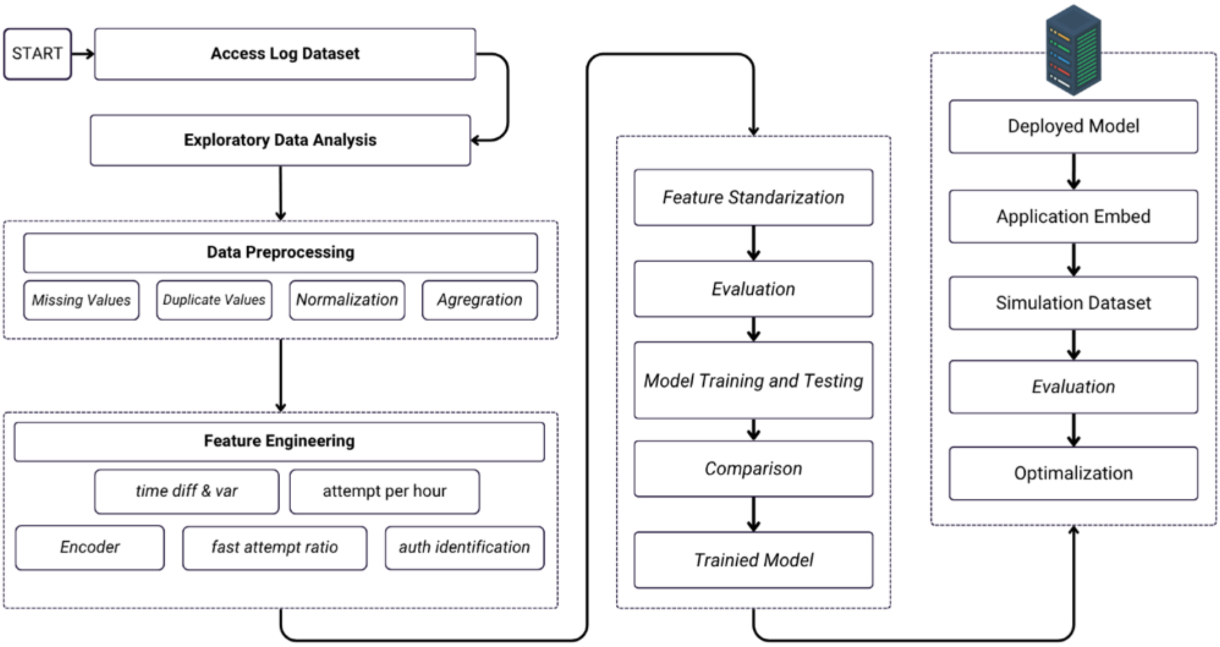

Figure 1. Research Method Flow for Brute-Force Attack Detection

This study employs an experimental design, which is chosen because normal data is relatively easy to obtain in large quantities, whereas attack data is rarely available and difficult to label [31]. Both IF and Rotated IF models were trained to learn normal behavioral patterns. To ensure that the data categorized as brute force, manual validation by cybersecurity experts was also conducted. This step ensured the accurate classification of brute-force instances. The models were subsequently evaluated on the combined testing dataset using standard evaluation metrics. This approach enables the models to detect anomalies without requiring attack labels during training. Early experiments were designed according to cross-validation principles to ensure model generalization and minimize the risk of overfitting. This evaluation methodology provides a robust framework for assessing the detection performance in realistic operational scenarios.

2.1. Dataset

A total of 569 log files were converted into a data frame, resulting in 25.371.532 rows of log before data preprocessing. After preprocessing, the dataset was reduced to 6.958.042 unique entries. The dataset was further enriched through feature engineering to capture the behavioral characteristics of access per IP address and sensitive paths, particularly those related to login or authentication processes. This feature engineering step involved generating new variables such as the number of login attempts per IP, frequency of access to authentication endpoints, and temporal patterns of requests. In addition, suspicious access patterns, such as repeated failed login attempts and unusually high request rates from a single IP address, were flagged for further analysis. These enhancements provided a comprehensive representation of access behaviors, enabling the detection of brute-force attacks in the dataset. The primary variables extracted from the log entries are presented in Table 1.

Table 1. Description of Each Variable in the Dataset

Variable	Description
filename	Name of the source log file.
ip	IP address of the client making the request.
time	Timestamp of the request.
method	HTTP method used
path	Requested URL path.
protocol	HTTP protocol version
status	HTTP status code
size	Response size in bytes.
referrer	Referring URL.
agent	Client’s User-Agent string.

2.2. Data Preprocessing

The data preprocessing phase commenced with log parsing, during which raw log entries were extracted using regular expressions, yielding 25,371,532 records. Subsequent cleaning procedures involved removing missing and duplicate entries; no missing values were identified, and 18,413,490 duplicates were eliminated, yielding 6,958,042 unique records. **To perform preprocessing and model training, the researchers used Google Colab with the following specifications: T4 GPU, 12 GB RAM, and 120 GB disk storage.** Thereafter, domain extraction and IP address normalization were conducted to ensure consistent formatting. The timestamp attribute was standardized to the %d/%b/%Y:%H:%M:%S %z format and appropriately renamed. Finally, user agent parsing was performed to derive information on operating systems and browsers, enriching the dataset with additional analytical features. The complete preprocessing workflow is depicted in Figure 2.

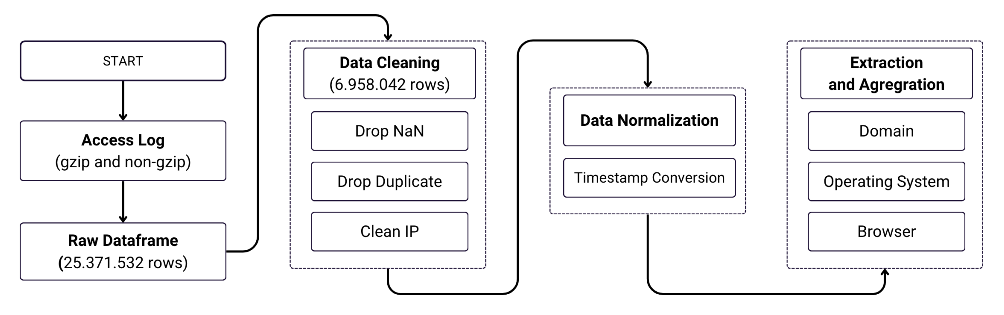

Figure 2. Data Preprocessing Stages of the Research

2.3. Feature Engineering Variables

This step aims to generate features that reveal brute-force attacks by examining request patterns per IP–path pair, since such attacks typically involve rapid, repeated requests to the same endpoint. Temporal features, such as the hour and day of the request, are extracted to detect timing-related attack patterns. Time intervals between consecutive requests from the same IP–path are then calculated, with both variance and standard deviation assessed; low values here can indicate regular, automated attack behaviour. Request frequency was calculated per IP–path pair, along with the average number of attempts per hour to measure attack intensity. The fast attempt ratio indicates how many requests in a session occur within 10 seconds, flagging brute-force activity when high. A binary feature marks login-related paths to highlight likely targets. The feature-engineered dataset supported the anomaly detection model's training and testing, and the resulting variables are shown in Table 2.

Table 2. Variables Generated from Feature Engineering

	Feature	Description
1	hour	For identifying attack patterns occurring at specific hours
2	day_of_week	Extracts the day of the week for detecting attacks on specific days.
3	ip_freq	Calculates the total number of logins attempts from each IP address.
4	time_diff	The time difference between consecutive login attempts from the same IP.
5	time_var	Low variance suggests consistent intervals, typical of automated attacks.
7	time_std	Low time_diff std dev indicates automated login activity.
8	attempts_per_hour	Calculates the avg number of logins attempts to measure attack intensity.
9	fast_attempt_ratio	Indicates the percentage of login attempts from a single IP.

2.4. Feature Standardization, Model Training, and Model Evaluation

RIF introduces a rotation process on the data before constructing the isolation trees. By applying rotation, the model can identify anomalies that might otherwise remain hidden or undetected in the original data dimensions, especially when there is a high correlation among features [20]. This rotation is performed randomly before each tree is built, allowing the model to capture complex anomaly patterns better. In general, RIF offers improved anomaly detection performance on high-dimensional or highly correlated datasets compared to the classical IF approach [32]. The model was initialized with several hyperparameters, including n_estimators, contamination, max_samples, random_state, and n_jobs. The n_estimators parameter determines the number of trees in the forest; a higher value generally leads to a more stable model. The contamination parameter in IF specifies the expected proportion of anomalies in the training data. This parameter influences the internal anomaly score threshold that the model uses to classify data as normal or anomalous. During the fitting process, IF constructs a series of isolation trees to learn the patterns of normal data.

2.5. Model Optimization and Comparison

After the IF and RIF models were trained on data categorized as normal, the evaluation phase assessed how well both models could detect brute-force attacks and correctly classify normal data on previously unseen samples. Model performance was measured using several metrics. Overall accuracy was calculated to determine the percentage of all data correctly classified. Benign classification accuracy measured how effectively the model recognized normal data as normal. Brute-force detection rate (recall) is the proportion of brute-force attacks the model successfully detects. Additional metrics included precision, F1-Score, and AUC, which reflect the model's overall ability to distinguish between positive and negative classes. This study directly compared standard IF, which uses axis-aligned data splits, with RIF, a variant that applies random rotations to detect off-axis anomalies better. To allow model reuse without retraining, the trained IF model, StandardScaler, IP encoder, feature names, and feature medians were saved, ensuring consistent preprocessing and prediction for new data.

3. RESULT AND ANALYSIS

3.1. Dataset Characteristics, Preprocessing, and Exploration

The behaviour-based feature engineering phase was applied to each IP–path pair. It included components such as numerical IP representation, request counts, timing variations, login attempt rates, temporal context, and identification of login paths. Feature exploration involved analysing distributions, correlations, and relevance to behavioural traits, while feature analysis identified the most significant predictors for anomaly detection. Figure 3 shows the top 10 IP addresses by frequency, which indicates that the dataset contains brute force attacks.

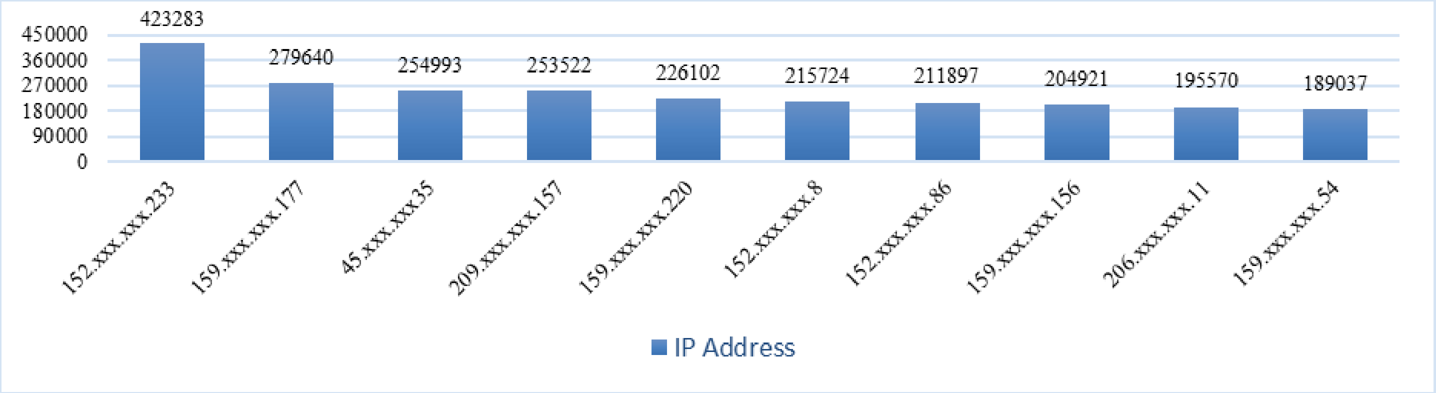

Figure 3. Distribution of the Top 10 Most Accessed IP Addresses

In addition to the IP Address, the greater number of POST methods compared to GET requests serves as one indication of rapid, intensive data transmission. Under normal circumstances, the amount of data sent should not significantly exceed the amount of data requested. The high volume of POST requests is a characteristic of brute force, which operates by repeatedly submitting credential guesses until a valid combination is found Figure 4.

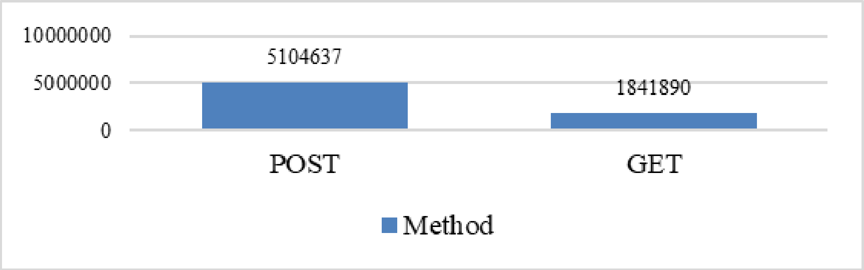

Figure 4. Distribution of HTTP Method Usage in Network Traffic

This study also computes temporal login features and detects IP–path pairs with high activity as potential brute-force indicators. The results are divided into normal data, attack data, and aggregate statistics, as shown in Figure 5. This study also computes temporal login features and detects IP–path pairs with high activity as potential brute-force indicators. The results are divided into normal data, attack data, and aggregate statistics, as shown in Figure 5. Table 3 presents the results of the temporal statistical features used to analyze brute-force patterns in activity logs. Features such as the average time between attempts, time variance, and fast attempt ratio serve as key indicators for distinguishing attack behaviours from normal activities.

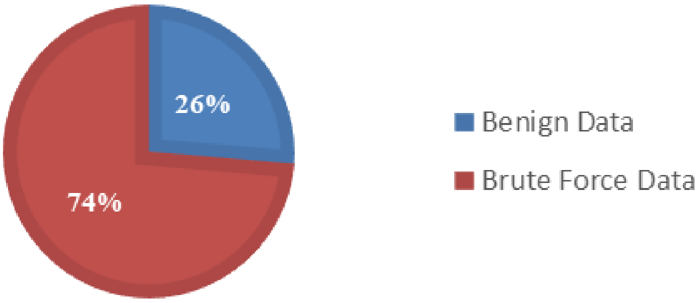

Figure 5. Proportion of Benign and Brute-Force Data in the Dataset

Table 3. Pembagian data untuk Training dan Testing

Statistic	Total Attempts	Avg Time Diff	Min Time Diff	Fast Attempts	Time Variance	Time Span Hours	Attempts Per Hour	Fast Attempt Ratio
count	85.65	25.111	25.111	85.650	16.468	85.650	85.650	85.650
mean	61.94	165.029	107.405	58.89	$4.15{\times}10^{10}$	36.85	36.25	0.0249
std	2.293	278.950	279.564	2.284.	$1.61{\times}10^{11}$	99.99	116.75	0.1150
min	1.0	0.0	0.0	0.0	0.0	0.0	0.0014	0.0
max	210	5.151	5.151	210	$5.62{\times}10^{12}$	1.430	8.775	0.999992

3.2. Model Performance

The IF model was built with 175 estimators, 0.1 contamination, auto max samples, full features, no bootstrap, and random_state=6, trained on standardized data. Both Standard IF and RIF were then evaluated on a mixed dataset of benign and brute-force attack data. Results show high accuracy for both models: RIF is slightly better at identifying benign traffic, while Standard IF had higher overall accuracy in Figure 6.

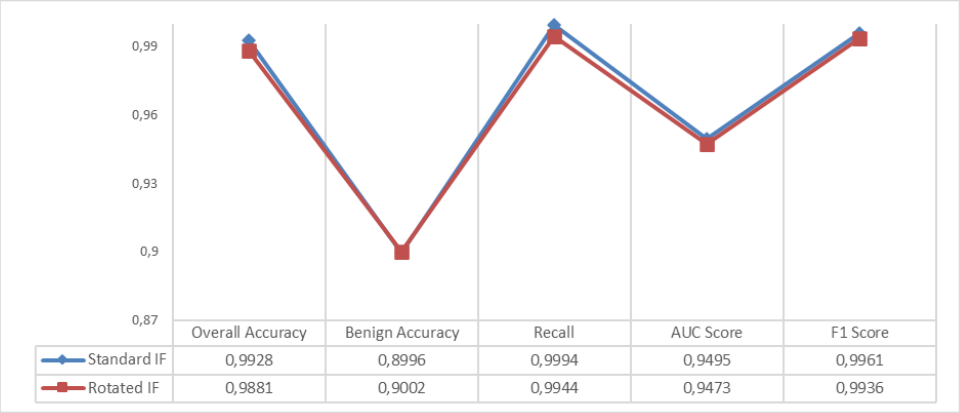

Figure 6. Performance Comparison Between the Standard IF and RIF Models

In terms of real-time capability, the developed prediction function, after loading the trained model and preprocessing components, was able to process new log data and efficiently generate detection statuses. Based on benchmarking, a prediction on an attack simulation data test set containing 100.000 records was completed quickly. The deployment model was able to scan the entire attack simulation dataset in 2.25 seconds, with an average time per item of 0.000023 seconds. Table 4 presents the performance evaluation results of the IF and RIF models in detecting brute-force attacks.

Table 4. Evaluation Metrics of Standard and Rotated IF Based on Benign and Brute Force Datasets

	Standard IF			**Rotated IF**		
	Precision	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
Benign	0.9915	0.8996	0.9433	0.920117	0.900223	0.910062
Brute Force	0.9929	0.9994	0.9961	0.992879	0.994414	0.993646
Accuracy	0.9928	0.9928	0.9928	0.988130	0.988130	0.988130
Macro avg	0.9922	0.9495	0.9697	0.956498	0.947318	0.951854
Weighted avg	0.9928	0.9928	0.9926	0.988025	0.988130	0.988070

As shown in Figure 7, the confusion matrix indicates that the IF model performed slightly worse, particularly in the Precision and F1-Score metrics for the Benign class. Nevertheless, the model still maintained a high detection rate against brute-force attacks. However, the overall Accuracy, Macro Average, and Weighted Average values decreased slightly. These findings indicate that while RIF is effective in detecting attacks, the standard IF model provides more consistent and reliable results across all evaluation metrics. Thus, the standard IF can be considered a more appropriate choice when overall classification performance and the balance between benign and malicious activity detection are top priorities.

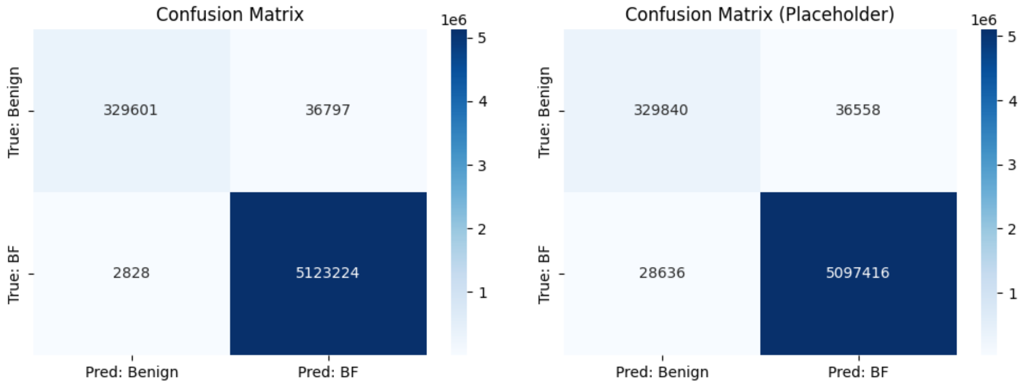

Figure 7. Confusion Matrix of the Standard and Rotated Model

3.3. Analysis of Model Evaluation Results

The IF model was trained on web access log data categorized as normal and evaluated on a test dataset that included both normal and brute force attack data. Based on the evaluation results, the model achieved a precision of 0.9929, indicating that the majority of its attack predictions were correct. The model's recall reached 0.9944, demonstrating its strong ability to identify brute-force attacks, with very few undetected intrusions accurately. Furthermore, the specificity value of 0.9002 reflects that the model performs reasonably well in recognizing normal activities, although some benign activities are still misclassified as attacks. These results indicate that the IF model effectively distinguishes between normal and malicious activities. This capability is highly valuable in cybersecurity contexts, as accurate attack detection helps prevent losses from brute-force intrusions. In addition, the high precision and recall values suggest that the model is highly effective for real-world deployment, where minimizing detection errors is essential. However, since the specificity is slightly below 1.00, there remains room for improvement, particularly in reducing false positives that may lead to false alarms and increase the workload of security teams. The evaluation also demonstrates that the IF model is highly sensitive to attack patterns, making it unlikely that brute-force attempts will go undetected. Consequently, this model is highly suitable for use as an early-detection system within web infrastructures vulnerable to brute-force threats.

Testing on diverse data containing various attack and normal activity patterns further reinforces the reliability of the IF model. The model also shows potential for further enhancement through additional feature engineering or parameter tuning to improve specificity. Compared to other approaches in the cybersecurity domain, IF offers a balanced trade-off between accurate attack detection and minimizing the misclassification of normal activities. From an implementation standpoint, the IF model is relatively easy to integrate into existing monitoring systems, as it does not require labelled attack data during training; only normal activity data is required. This characteristic reduces data preparation efforts and accelerates deployment. Overall, the evaluation results confirm that the IF model is a promising solution for detecting brute-force attacks in modern web applications. This model achieved an accuracy of 0.9881, with a recall of 0.9944 and a precision of 0.9929, indicating strong performance in identifying attack activities. The specificity value of 0.9002 indicates that the model performs fairly well at recognizing benign traffic, though it is slightly lower than that of the IF model. Overall, RIF maintains robust, consistent performance, with evaluation metrics closely comparable to those of the standard model.

The differences in accuracy and recall are small. Given the very large sample size, even small differences can be statistically significant, but the document notes that the performance gap is practically insignificant. This is because the engineered features already provide strong separation between normal and anomalous data, and rotation does not reveal additional anomaly structures in this context.

3.4. Interpretation of Key Metrics

The key metrics for evaluating the brute-force attack detection model provide a comprehensive understanding of its strengths and weaknesses in distinguishing between normal activities and attacks. The high precision value (0.9929) indicates that most of the attacks detected by the model are indeed actual attacks, thereby minimizing the risk of false positives. The very high recall (0.9944) confirms that the model is almost always able to detect brute-force attacks, making the likelihood of missed detections very low. Interpreting this confusion matrix is essential for understanding the model's error patterns and enabling targeted improvements, such as parameter tuning, to increase specificity without compromising sensitivity. Thus, the IF model is not only effective in detecting attacks but also reliable in minimizing false alarms that could otherwise burden the security team. The interpretation of the key metric is shown in Table 5.

Table 5. Performance Evaluation Results of the Brute Force Attack Detection Model

Metric	Value	Interpretation
True Positive (TP)	5.123.224	Number of brute force attacks detected. A high value indicates the model is highly effective at recognizing malicious activity.
False Negative (FN)	2.828	Instances of brute-force attacks that the model missed. A low number indicates a very low attack evasion rate.
True Negative (TN)	329.601	Number of normal (benign) activities correctly classified as safe. This reflects the model's ability to recognize normal traffic.
False Positive (FP)	36,797	Normal activities are incorrectly classified as attacks. This value should be controlled to prevent false alarms.
Recall (Brute Force Detection Rate)	99.94% (0.9994)	Percentage of actual brute force attacks successfully detected. A near-perfect value indicates extremely high model sensitivity.
Precision (Brute Force)	99.29% (0.9929)	Percentage of predicted attacks that are actual attacks. A high value shows the model rarely issues false alerts.
Specificity (True Negative Rate)	89.96% (0.8996)	The model's ability to correctly identify normal activities. Some benign activities are still misclassified as attacks.
Benign Classification Accuracy	89.96% (0.8996)	Accuracy in recognizing normal activity is used to evaluate the balance between attack detection and normal traffic.
F1-Score	0.9961	The harmonic mean of Precision and Recall. A very high value indicates an optimal balance between accurate detection and minimal errors.
AUC (Area Under Curve)	0.949510	Indicates the model's ability to distinguish between attacks and normal traffic. A value close to 1 demonstrates excellent classification performance.
Accuracy (Overall)	0.992786 (99.28%)	Overall percentage of correct predictions. A near-perfect value shows the model's strong general performance.

Overall, these metrics indicate that the developed IF model is highly effective at detecting brute-force attacks in this web access log dataset. The high detection rate is a primary strength, crucial in cybersecurity applications to minimize undetected attacks. The relatively controlled False Positive rate is notable given the data volume and the complexity of traffic patterns. Several factors support the effectiveness of IF for brute force detection in web access logs: IF operates by isolating outliers rather than modeling normal data. Anomalies, such as repeated login attempts from the same IP within a short interval, are inherently easier and faster to isolate within a random tree structure compared to normal data, which tends to cluster. Brute force patterns are intrinsically ”isolated” from typical user behavior. IF exhibits better time complexity than certain other anomaly detection algorithms when applied to high-dimensional, large-volume datasets, which is particularly relevant for web access logs. IF does not require labeled attack data for training. This is a significant advantage, as accurately obtaining labeled attack data can be highly challenging and time-consuming in real-world scenarios.

3.5. Role of Feature Engineering

The feature engineering stage plays a crucial role in the model's success. Engineered features, such as frequency per IP-path pair, time differences between requests, fast attempt ratio, and login path indicator, directly capture behavioural characteristics that distinguish brute force attacks from normal traffic. For example, brute force attacks tend to exhibit high ip_path_freq and attempts_per_hour, low time_var and time_std, a high fast_attempt_ratio, and is_login_path set to True. These features enable IF to isolate such anomalous patterns effectively. Feature standardization using StandardScaler ensures that all features contribute proportionally to the isolation process, preventing features with large value ranges from disproportionately dominating. Consequently, feature engineering not only improves model performance but also helps reduce false positives, as anomalous patterns become more clearly separated from normal traffic. Additionally, selecting relevant features accelerates model training by focusing only on important attributes. This process also opens the door to identifying new attack patterns in the future by introducing innovative features as needed. Domain knowledge is particularly important at this stage to ensure that engineered features accurately reflect real attack characteristics. With well-developed features, the IF model can be adapted to detect various types of attacks beyond brute force through appropriate feature adjustments.

3.6. Comparison with Other Approaches

This study demonstrates that IF offers significant advantages over traditional detection methods. Test results indicate that IF can detect attack patterns with high accuracy and efficiency, while also being more adaptive to the continuously evolving dynamics of cyber threats. To clarify the contribution and positioning of this algorithm, a comparative analysis was conducted against several commonly used intrusion detection system approaches.

Traditional methods, such as rule-based or threshold-based detection, remain widely used due to their simplicity and ease of implementation. These systems typically operate based on explicit rules—for example, blocking an IP address if the number of failed login attempts exceeds a certain threshold within a specified time frame [33]. However, these approaches have several fundamental limitations. First, they are static and non-adaptive, making it difficult to accommodate new attack patterns. Second, their sensitivity to thresholds makes them prone to misclassification: False Positives may restrict legitimate users, while False Negatives allow actual attacks to go undetected. Third, these methods are easily circumvented by attackers who adjust the speed or pattern of their attacks to avoid detection thresholds. These conditions indicate that rule-based methods are no longer sufficient to address increasingly dynamic and complex attack patterns.

IF algorithm offers a more adaptive, data-driven approach. Unlike traditional methods [9] that rely on fixed rules, IF automatically learns the system's normal behavior and identifies significant deviations as anomalies. This allows the model to detect unusual activities, such as sudden spikes in login attempts from a single IP, without relying on manually defined thresholds. Its adaptive capabilities make IF more effective at detecting new attacks or variants that lack signatures. Compared to signature-based detection, IF demonstrates superior performance. Signature-based approaches detect attacks by matching known attack patterns, relying on an up-to-date signature database [34–36]. Consequently, these systems cannot detect new or previously unknown attack variants, making them inherently reactive. In contrast, IF is proactive, focusing on identifying abnormal behavior rather than matching specific patterns. In other words, IF does not search for known attacks but detects deviations from normal data behavior, providing early warnings for attacks without established signatures.

Beyond traditional and signature-based methods, unsupervised anomaly detection algorithms such as One-Class Support Vector Machine (OCSVM) and Local Outlier Factor (LOF) are also commonly used. However, both approaches face notable limitations when applied to large-scale log data. OCSVM, as a distance-based algorithm, suffers from high computational complexity, making it inefficient for large or high-dimensional datasets, as its training and inference processes require substantially more time and resources. LOF performs well at detecting local anomalies in small to medium-sized datasets, but its effectiveness decreases significantly when handling large, heterogeneous log data. In contrast, Isolation Forest (IF) offers superior efficiency due to its linear computational complexity and its ability to rapidly isolate anomalies through random partitioning without relying on costly distance calculations. The results of this study further reinforce the advantages of IF in anomaly detection: it demonstrates exceptional efficiency when processing large-scale web access logs—here involving 1,831,989 benign and 5,126,052 brute-force entries; it achieves strong performance metrics with a recall of 99.94%, precision of 99.29%, F1-score of 99.61%, and accuracy of 99.28%; and it can be trained solely on normal data, removing the need for labeled attack samples, which are often scarce in real-world cybersecurity environments.

3.7. Model Optimization

The model optimization process in this study focused on improving the accuracy of brute-force attack detection through two main approaches: parameter tuning of the Isolation Forest (IF) algorithm and exploration of the Refined Isolation Forest (RIF) method. Both approaches were developed using real web access log data collected from several server domains targeted by brute-force attacks against the MariaDB authentication system. The IF model was trained on standardized, benign data to ensure equal feature scales, preventing any single feature from dominating during isolation. Data preprocessing with StandardScaler plays a crucial role in the optimization process, ensuring each feature has a mean of zero and a standard deviation of one. This normalization effectively reduces bias against features with a wide range of values, allowing each feature to contribute proportionally to the distance calculation during the isolation process. Furthermore, parameter tuning was conducted experimentally by adjusting model complexity to the dataset's size and variability until the optimal configuration was achieved. The final parameters used in this experiment are summarized in Table 6.

Table 6. Optimal IF Parameters Used for Attack Detection

Parameter	Value	Description
n_estimators	175	Number of trees used to build the IF ensemble, providing a balance between prediction stability and computational time.
contamination	0.1	Assumes approximately 10% of the data are anomalies, consistent with the proportion of attacks in the preprocessed dataset.
max_samples	'auto'	Each tree is trained on a random subset of the data to maintain isolation diversity.
max_features	1.0	All features are used to capture IP behavior dimensions fully.
bootstrap	FALSE	The model does not use repeated sampling, improving memory efficiency.
random_state	6	Ensures reproducibility of experimental results.

In Figure 8, this study tested the Refined Isolation Forest (RIF) approach using two feature rotation schemes. The first scheme applies Principal Component Analysis (PCA)-based rotation. At this stage, a rotation matrix is derived from the principal components of the training data via a full PCA decomposition to ensure optimal feature representation. PCA-based rotation aims to transform the original feature space into a new, more compact space, enabling more efficient anomaly isolation. This approach is expected to improve the model's ability to accurately separate benign and anomalous data patterns.

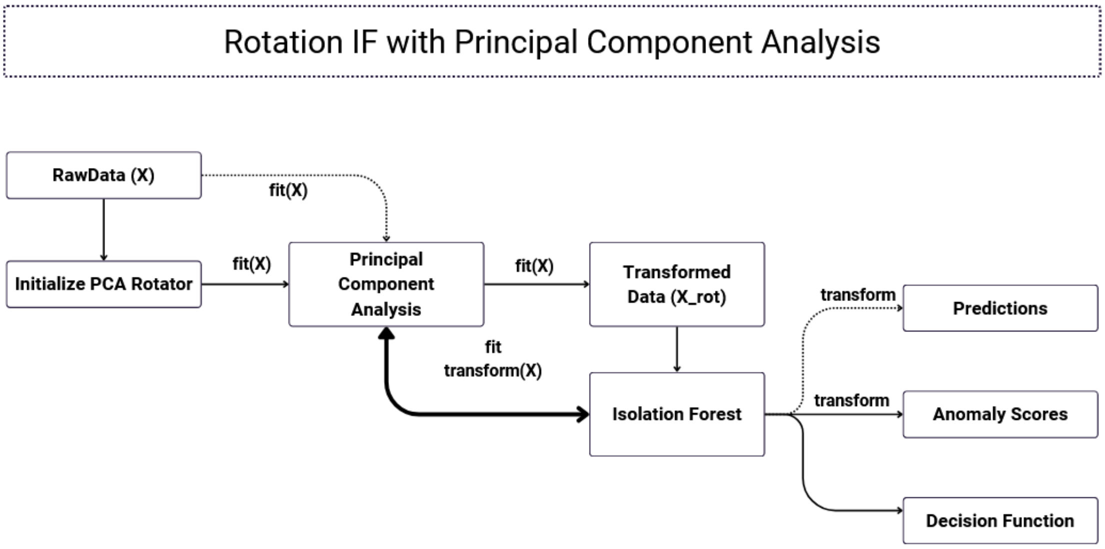

Figure 8. RIF Using Principal Component Analysis (PCA)

The PCA component matrix serves as an orthogonal transformation that rotates the feature space along directions of maximum variance. Mathematically, if the original data is $X \epsilon R^{n \times d}$ and the PCA rotation matrix is $R_i \epsilon R^{d \times d}$, the rotated data is obtained as shown in Formula 1.

$$X_{rotated} = X R_i^T \tag{1}$$

Since R_i is orthogonal ($R_i^T R_i = I_d$), this transformation preserves the Euclidean distances between data points. This allows anomalous patterns hidden in the original orientation to be more easily detected by the RIF model. Each IF instance in the ensemble is trained with a single tree (n_estimators=1) on the rotated data, enabling diversification through rotation rather than data or feature subsampling as in the standard implementation. Second, Gaussian Random Projection (GRP), as shown in Figure 9, was tested as an alternative rotation scheme. GRP randomly rotates the feature space using a Gaussian rotation matrix, enabling isotropic exploration. Random rotations can capture anomalous patterns that may remain hidden under both the original orientation and PCA rotation.

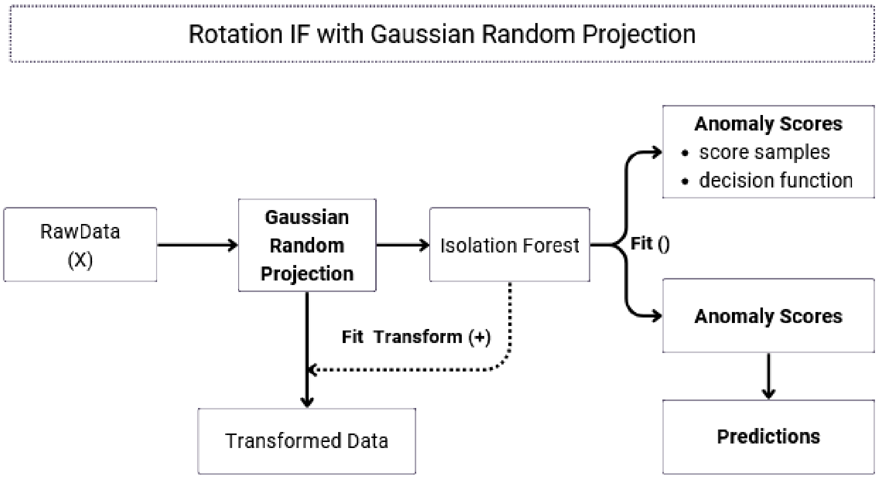

Figure 9. RIF Using Gaussian Random Projection

Evaluation showed that GRP achieved superior detection performance. Therefore, the final RIF implementation employs GRP as the main rotation stage prior to isolation tree construction. This approach combines ensemble learning principles with geometric transformations in the feature space, enhancing flexibility in detecting anomalies across diverse data distributions. Table 7 shows a significant difference between IF and RIF.

Table 7. Significant Differences Between IF and RIF

Aspect	Standard Isolation Forest (IF)	Rotated Isolation Forest (RIF)	Remarks / Implications
Working Principle	Performs random splits on each feature (axis-aligned). Anomalies that are far from normal clusters become isolated more quickly.	Rotates the feature space (using PCA or Gaussian Random Projection) before building isolation trees, allowing splits to occur along new orientations rather than only the original axes.	Rotation aims to detect hidden anomalies that may not be visible in the original axis-aligned space.
Theoretical Difference	More efficient and optimal when the engineered features can already separate normal and anomalous data clearly.	Theoretically superior for high-dimensional data with strongly correlated features, as rotation can uncover anomaly patterns missed by axis-aligned splits.	The theoretical advantage of RIF does not appear when the dataset has low dimensionality and weak feature correlations.
Experimental Results	Recall: 99.94% Precision: 99.29% F1-Score: 99.61% Accuracy: 99.28% AUC: 0.9495	Recall: 99.44% Precision: 99.29% F1-Score: 99.36% Accuracy: 98.81% AUC: 0.9473	Standard IF slightly outperforms RIF in recall, F1-score, and accuracy. Rotation does not yield a meaningful improvement.
Explanation of Non-	- Features are highly discriminative (e.g., login frequency, time intervals, fast attempt ratios).	- Feature rotation adds no new information.	
Significant Difference	- Data dimensionality is moderate and feature correlations are weak. - More efficient in computation and easier to deploy in production.	- Does not reveal additional anomaly structures since existing features already provide strong separation.	Both models perform excellently; however, the performance gap is statistically insignificant.
Conclusion	Practically more effective for brute-force detection on MariaDB due to already well-engineered features.	Does not provide a significant performance gain on this dataset.	A noticeable difference would only emerge in high-dimensional, highly correlated, or more complex datasets.

Experimental results indicate that IF parameter tuning yields very high performance, with a Recall of 99.94%, F1-Score of 99.61%, and overall Accuracy of 99.28%. In contrast, the RIF model with random projection achieved a Recall of 99.44% and an Accuracy of 98.81%, slightly lower than the standard IF. Detailed analysis shows that the temporal and frequency features used are already highly discriminative, so the benefit of random rotation is not yet significant. Another possibility is that the Gaussian transformation does not optimally map the feature space to orientations that facilitate anomaly isolation. The comparative results contribute theoretically by clarifying that the practical effectiveness of feature rotation in IF models depends on the nature of the dataset. For well-engineered, moderately dimensional data, standard IF is sufficient and more efficient. The study thus refines the understanding of IF model selection and optimization in cybersecurity anomaly detection.

3.8. Limitations and Areas for Improvement

Despite achieving high detection performance, the models developed in this study still face several important limitations that warrant careful consideration. The effectiveness of both IF and RIF is strongly influenced by hyperparameter tuning, particularly the number of estimators and the contamination rate, and suboptimal configurations can disrupt the balance between precision and recall, ultimately reducing detection quality. Model reliability is also shaped by the quality of data labeling and manual validation, as heuristic labeling practices and inconsistent data quality may introduce bias into both the training and evaluation processes. Another crucial challenge lies in determining appropriate anomaly score thresholds; thresholds that are set too high or too low can lead to excessive false positives or false negatives, limiting the practical applicability of the models in operational environments. Interpreting these numerical scores demands sufficient domain knowledge to ensure that decisions align with broader security policies and organizational needs. Furthermore, because the study is based on MariaDB log data with feature engineering tailored specifically to that environment, the findings may not be directly transferable to other database platforms or network settings without additional validation. Although the IF model performs strongly on the evaluated dataset, its resilience against future, evolving attack patterns and novel threats remains uncertain, suggesting the need for future work on adaptive feature design and dynamic thresholding strategies.

Finally, the study underscores the necessity for stricter validation of heuristic labels and more rigorous data validation procedures, as limited validation may undermine scientific rigor and reduce the reproducibility of the research.

4. CONCLUSION

Based on the experimental results and discussions presented, it can be concluded that IF parameter optimization delivers very high performance in brute-force attack detection, as reflected by a Recall of 99.94%, an F1-Score of 99.61%, and an overall Accuracy of 99.28%. These values demonstrate that the IF model almost always identifies brute force patterns with excellent sensitivity to normal data.

Meanwhile, the RIF approach incorporating Gaussian Random Projection achieved a Recall of 99.44% and an Accuracy of 98.81%, slightly lower than those of the standard IF model. The addition of random rotation in RIF shows potential for altering the feature space structure but has not significantly enhanced detection performance in this study. This suggests that the temporal and frequency features used are already highly effective in discriminating between normal and anomalous activity.

Nevertheless, the study also highlights important limitations, including the need for stricter validation of heuristic labels, the model's sensitivity to the quality of the training data, and the importance of selecting anomaly score thresholds aligned with business and security contexts. Therefore, developing an effective anomaly detection system requires a systematic approach, layered validation, and deep domain understanding to ensure detection results can be optimally integrated into organizational security policies. Recommendations for future research include exploring more robust data validation techniques, developing more adaptive features, and implementing dynamic thresholds based on anomaly score distributions to enhance overall system reliability.

5. ACKNOWLEDGEMENTS

The authors would like to express their gratitude to previous researchers [20, 28, 36, 30] whose works on the Standard Isolation Forest and Rotated Isolation Forest provided the essential foundation for this study.

6. DECLARATIONS

AUTHOR CONTIBUTION

Hartono designed the study, feature engineering, performed the data analysis, and drafted the main manuscript. Khusnul Khotimah contributed to data processing, manual validation, and literature review. Rokin Maharjan provided input on the methodology, literature review, model evaluation, and manuscript revision. All authors have read and approved the final version of this article.

FUNDING STATEMENT

This research did not receive any specific grant from public, commercial, or not-for-profit funding agencies. All research expenses were covered independently by the authors.

COMPETING INTEREST

The authors declare that there are no competing interests associated with this research.

REFERENCES

[1] I. M. Lina and G. R. Fernandes, "Anticipate password security with burp suite using the brute force attack method," vol. 7, no. 1, pp. 118–127, June,2023, https://doi.org/10.37339/e-komtek.v7i1.1162.

[2] N. Alaa and F. Al-Shareefi, "A comparative study between two cybersecurity attacks: Brute force and dictionary attacks," vol. 11, no. 2, pp. 133–139, 2024, https://doi.org/10.31642/JoKMC/2018/110216.

[3] Y. Wu, P. M. Cao, A. Withers, Z. T. Kalbarczyk, and R. K. Iyer, "Mining threat intelligence from billion-scale SSH brute-force attacks," in *Proceedings 2020 Workshop on Decentralized IoT Systems and Security*. Internet Society, 2020, https://doi.org/10.14722/diss.2020.23007.

[4] B. Pal *et al.*, "Might I get pwned: A second generation compromised credential checking service," in *31st USENIX Security Symposium (USENIX Security 22)*, 2022, pp. 1831–1848. [Online]. Available: https://www.usenix.org/conference/usenixsecurity22/presentation/pal

[5] M. H. Nguyen Ba, J. Bennett, M. Gallagher, and S. Bhunia, "A case study of credential stuffing attack: Canva data breach," in *2021 International Conference on Computational Science and Computational Intelligence (CSCI)*, 2021, pp. 735–740, https://doi.org/10.1109/CSCI54926.2021.00187.

[6] N. Hubballi, N. Tiwari, and P. Khandait, "POSTER: Distributed SSH bruteforce attack detection with flow content similarity and login failure reputation," in *15th ACM Asia Conference on Computer and Communications Security*, 2020, pp. 916–918, https://doi.org/10.1145/3320269.3405443.

[7] N. Tiwari and N. Hubballi, "Secure socket shell bruteforce attack detection with petri net modeling," vol. 20, no. 1, pp. 697–710, 2023-03, https://doi.org/10.1109/TNSM.2022.3212591.

[8] F. Wilkens and M. Fischer, "Towards data-driven characterization of brute-force attackers," in *2020 IEEE Conf. Commun. Netw. Secur. CNS*, 2020, pp. 1–9, https://doi.org/10.1109/CNS48642.2020.9162326.

[9] G. Fahrnberger, "Pattern-and similarity-based realtime risk monitoring of SSH brute force attacks with bloom filters," in *2024 36th Conf. Open Innov. Assoc. FRUCT*, 2024, pp. 133–144, https://doi.org/10.23919/FRUCT64283.2024.10749895.

[10] A. Raj *et al.*, "Brute forcing on secured shell servers emphasising the role of cyber forensics – a quali-quantitative study," vol. 92, no. 3, pp. 152–157, September,2024, https://doi.org/10.1177/00258172241236269.

[11] D. Stiawan, g.-i. family=Idris, given=Mohd. Y., R. F. Malik, S. Nurmaini, N. Alsharif, and R. Budiarto, "Investigating brute force attack patterns in IoT network," vol. 2019, no. 1, p. 4568368, 2019, https://doi.org/10.1155/2019/4568368.

[12] A. Subhan, Y. N. Kunang, and I. Z. Yadi, "Analyzing the attack pattern of brute force attack on SSH port," pp. 67–72, 2023, https://doi.org/10.1109/ICITCOM60176.2023.10441929.

[13] O. Mykhaylova, A. Shtypka, and T. Fedynyshyn, "An Isolation Forest-based approach for brute force attack detection," in *1st International Workshop on Bioinformatics and Applied Information Technologies (BAIT 2024)*, 2024, pp. 43–54. [Online]. Available: https://ceur-ws.org/Vol-3842

[14] M. Elnour, N. Meskin, K. Khan, and R. Jain, "A dual-isolation-forests-based attack detection framework for industrial control systems," vol. 8, pp. 36 639–36 651, 2020, https://doi.org/10.1109/ACCESS.2020.2975066.

[15] H. Xu, G. Pang, Y. Wang, and Y. Wang, "Deep isolation forest for anomaly detection," vol. 35, no. 12, pp. 12 591–12 604, December,2023, https://doi.org/10.1109/TKDE.2023.3270293.

[16] G. Pang, C. Shen, L. Cao, and A. Van Den Hengel, "Deep learning for anomaly detection: A review," vol. 54, no. 2, pp. 38:1–38:38, March,2021, https://doi.org/10.1145/3439950.

[17] G. Pu, L. Wang, J. Shen, and F. Dong, "A hybrid unsupervised clustering-based anomaly detection method," vol. 26, no. 2, pp. 146–153, April,2021, https://doi.org/10.26599/TST.2019.9010051.

[18] L. Ruff *et al.*, "A unifying review of deep and shallow anomaly detection," vol. 109, no. 5, pp. 756–795, May,2021, https://doi.org/10.1109/JPROC.2021.3052449.

[19] V. Monemizadeh and K. Kiani, "Detecting anomalies using rotated isolation forest," vol. abs/2501.17787, 2025, https://doi.org/10.48550/arXiv.2501.17787.

[20] ——, "Detecting anomalies using rotated isolation forest," vol. 39, no. 3, p. 24, March,2025, https://doi.org/10.1007/s10618-025-01096-5.

[21] G.-P. Fernando, A. M. Florina, and C.-B. Liliana, "Evaluation of the performance of unsupervised learning algorithms for intrusion detection in unbalanced data environments," vol. 12, pp. 190 134–190 157, 2024, https://doi.org/10.1109/ACCESS.2024.3516615.

[22] M. Nalini, B. Yamini, C. Ambhika, and R. S. Subramanian, "Enhancing early attack detection: Novel hybrid density-based isolation forest for improved anomaly detection," vol. 16, no. 5, pp. 3429–3447, June,2025, https://doi.org/10.1007/s13042-024-02460-5.

[23] W. Chua *et al.*, "Web traffic anomaly detection using isolation forest," vol. 11, no. 4, p. 83, December,2024, https://doi.org/10.3390/informatics11040083.

[24] Y. Xu, H. Dong, M. Zhou, J. Xing, X. Li, and J. Yu, "Improved isolation forest algorithm for anomaly test data detection," vol. 9, no. 8, pp. 48–60, August,2021, https://doi.org/10.4236/jcc.2021.98004.

[25] L. Max, S. Florian, W. Markus, H. Wolfgang, and R. Andreas, “AIT log data set V1.1,” 2020, https://doi.org/10.5281/ZENODO.4264796.

[26] M. Hogan, Y. Michalevsky, and S. Eskandarian, “DBREACH: Stealing from databases using compression side channels,” in *2023 IEEE Symp. Secur. Priv. SP*, 2023, pp. 182–198, https://doi.org/10.1109/SP46215.2023.10179359.

[27] C. Rookard and A. Khojandi, “Unsupervised machine learning for cybersecurity anomaly detection in traditional and software-defined networking environments,” vol. 22, no. 2, pp. 1129–1144, April,2025, https://doi.org/10.1109/TNSM.2024.3490181.

[28] S. U. Shankari, H. Mohameed, M. Kulkarni, S. Aravindh, and N. Purushotham, “Cybersecurity threat detection in smart cities using box plot sampling isolation forest,” in *2025 Int. Conf. Intell. Syst. Comput. Netw. ICISCN*, 2025, pp. 1–5, https://doi.org/10.1109/ICISCN64258.2025.10934339.

[29] T. A. Almoabady *et al.*, “Protecting digital assets using an ontology based cyber situational awareness system,” vol. 7, 2025, https://doi.org/10.3389/frai.2024.1394363.

[30] J. A. Pawar, M. S. Avhankar, A. Gupta, A. Barve, H. Patil, and R. Maranan, “Enhancing network security: Leveraging isolation forest for malware detection,” in *2024 2nd International Conference on Advancement in Computation & Computer Technologies (InCACCT)*, 2024, pp. 230–234, https://doi.org/10.1109/InCACCT61598.2024.10550968.

[31] U. Bhadani, “Advanced email security with NLP and the isolation forest algorithm,” in *2024 IEEE 12th Int. Conf. Inf. Commun. Netw. ICICN*, 2024, pp. 497–503, https://doi.org/10.1109/ICICN62625.2024.10761702.

[32] J. Liang, H. Shui, R. Gupta, D. Upadhyay, and E. Darve, “Transfer learning for anomaly detection in rotating machinery using data-driven key order estimation,” vol. 22, pp. 13 310–13 326, 2025, https://doi.org/10.1109/TASE.2025.3552009.

[33] L. Wang *et al.*, “Incorporating gradients to rules: Towards lightweight, adaptive provenance-based intrusion detection,” 2024, https://doi.org/10.14722/ndss.2025.23822.

[34] M. Agoramoorthy, A. Ali, D. Sujatha, M. T. F. Raj, and G. Ramesh, “An analysis of signature-based components in hybrid intrusion detection systems,” in *2023 Intelligent Computing and Control for Engineering and Business Systems (ICCEBS)*, 2023, pp. 1–5, https://doi.org/10.1109/ICCEBS58601.2023.10449209.

[35] T. Sommestad, H. Holm, and D. Steinvall, “Variables influencing the effectiveness of signature-based network intrusion detection systems,” vol. 31, pp. 711–728, 2021, https://doi.org/10.1080/19393555.2021.1975853.

[36] U. Bhadani, “Advanced email security with NLP and the isolation forest algorithm,” in *2024 IEEE 12th International Conference on Information, Communication and Networks (ICICN)*, 2024, pp. 497–503, https://doi.org/10.1109/ICICN62625.2024.10761702.

[This page intentionally left blank.]