Cyber Attack Classification Using Random Forests in Online Learning System Network Infrastructure

Dwi Kurnia Wibowo[1]*, Agus Darmawan[2], Devi Astri Nawangnugraeni[3], Justicio Caesario [4]
[1,2,3,4]Department of Informatics, Engineering Faculty, Universitas Jenderal Soedirman, Banyumas, Indonesia

Article Info

Article history:

Received 12 24, 2025
Revised 01 04, 2026
Accepted 06 25, 2026

Keywords:

Attack Detection;
Machine Learning;
Random Forest;
Network Forensics;
Investigation.

ABSTRACT

Eldiru Unsoed and various other academic information systems are important digital assets that are susceptible to cyberattacks, and traditional rule-based Web application firewalls have detection flaws. It has been demonstrated that the typical ModSecurity system with Core Rule Set (CRS) only has a recall of 5.34%, meaning it misses most real assaults and leaves security holes. To address this issue, this paper creates a detecting system based on the Random Forest algorithm. The Eldiru Unsoed system's Nginx server logs from December 2024 to January 2025 provided the majority of the training data, which was then verified using the publicly available CSIC 2010 dataset. The model was created by developing hybrid features that incorporated lexical analysis, CRS rule context, and N-grams to categorize online traffic based on the log analysis. According to the assessment findings, the suggested Machine Learning-Random Forest (ML-RF) model increases F1-Score from 10.10% to 80.00% and recall from 5.34% to 72.00%. While keeping precision at 91.00%, this improvement in metrics shows that machine learning integration results in a more balanced and dependable cyber defense system to handle the difficulties of contemporary threat detection in safeguarding digital assets.

Corresponding Author:

Dwi Kurnia Wibowo,
Fakultas Teknik, Program Studi Informatika,
Universitas Jenderal Soedirman, Banyumas, Indonesia,
Email: dwi.kurnia@unsoed.ac.id

1 INTRODUCTION

The increasing frequency of network assaults necessitates research, comprehension, and advancement of more potent security defense systems. Every company, industry, and governmental level needs network security solutions to guard against the growing threat of cyberattacks. Network security requirements must be adjusted to the organization's current demands since no network is impervious to cyberattacks and there is a growing need for more dependable and efficient network security solutions to safeguard client and company data [1].

Several approaches have been put out to deal with and classify network traffic attacks. The first method is port-based, meaning that the Internet Assign Number Authority (IANA) must choose port numbers from a file. However, because of the growing number of applications and faulty ports, this approach has not worked. Moreover, applications that do not register their ports with IANA or that use dynamic port numbers are not covered by this technique. Another approach that has been proposed is the payload-based approach, also referred to as Deep Packet Inspection (DPI), which looks at and compares network packet contents with a database. Although it doesn't function with network apps that use encrypted data, this method yields more accurate results than port-based techniques [2], [3].

One of the most prevalent and harmful kinds of attacks is DDoS, which can stop authorized users from using network services. DDoS assaults can be used to target servers by overloading the network with traffic, which can deplete its resources. Additionally, the Internet of Things era has made a large number of gadgets capable of connecting to the Internet. As a result, attackers can conduct several DDoS assaults using a large number of bots from different places. It is challenging to detect DDoS attacks that are executed by bot devices. These attacks also rapidly deplete network resources. Serious DDoS assaults may damage a company's reputation and cost it up to £100,000 per hour. Channels for communication between the controller and the application layer or between the controller and the open flow switch are among the layers of SDN that might be overloaded by DDoS assaults. The entire network will shut down immediately if a DDoS assault destroys SDN since it has a single point of failure [4], [5], [6], [7].

Cyberattacks have also targeted web apps, which are an example of technical development. As online applications have become more widely developed and used, attacks on the web have become more frequent and more severe. In 2018, 953,000 online assaults were prevented daily, up to 611,000 the year before. According to the Open online Application Security Project (OWASP), injection vulnerabilities continue to be the most prevalent in online applications. Cross-Site Scripting (XSS) attacks are now classified as injection attacks, even in its most recent version [8], [9], [10], [11], [12].

The loss of integrity brought on by cyberattacks causes agencies to incur extra losses in the form of substantial recovery expenditures. Any action that damages system owners on computer networks, including disruption, data theft, and destructive activity, is prohibited and subject to legal action. Evidence discovered by network forensic techniques can be used to penalize offenders [13], [14].

Forensic investigations, which use IDS logs and attack notification systems, are commonly carried out by investigators using network monitoring systems like IDS. When suspicious behavior occurs on the network, intrusion detection systems (IDS) monitor it, alert users, and promptly report it as a warning. Digital signatures are the basis for the majority of intrusion detection systems. Many mistakes are made in detecting assaults as a result of the fluctuation in network traffic, which causes an increase in alerts because of the non-stationary nature of data flow in the network to create and respond to rising alerts [15], [16], [17].

Techniques for data mining and machine learning (ML) are crucial for identifying and categorizing cyberattacks. Developing methods to recognize and detect novel forms of assaults and helping investigators analyze evidence in network forensics may be possible using machine learning. Numerous studies on machine learning have been carried out in a variety of fields [18]. this method gives network devices anomaly-based intrusion detection capabilities [19], [20], [21], [22]. Due to machine learning's quick progress, several approaches with unique benefits and drawbacks have emerged that may be applied to a range of situations. Because it can clearly identify data points by constructing hyperplanes in n-dimensional space—where n is the number of features—Random Forest (RF) is one machine learning approach that may be utilized in classification.

The difficulties experienced by digital forensic investigators who have been looking into digital evidence using traditional techniques may be resolved with the introduction of machine learning. These difficulties include dispersion, heterogeneous data, and massive volumes of data that are too big for people to analyze rapidly [23], [24]. By helping investigators handle vast volumes of data more rapidly, this technology can help them spot issues

in accidents more rapidly and efficiently [25], [26], [27]. Through the use of Random Forest (RF) in attack data categorization, the research seeks to aid network forensic investigators in particular while conducting investigations and maintaining digital evidence.

2 RESEARCH METHOD

Two data sources were used in this work to train and assess the attack detection model. The access.log file of the Nginx web server hosting the Eldiru Unsoed system served as the main source of data for model training. The data was collected between December 1, 2024, and January 31, 2025. Ten million entries were chosen at random from a total of 77 million. After that, this sample was run via a replay simulation script to produce an audit.log file from ModSecurity. This file served as the final training data as it was enhanced with security context, including triggered rules and anomaly scores.

The ECML/PKDD 2007 Challenge and the CSIC 2010 Dataset are two well-known public datasets that are used in this work for testing, assessment, and data enrichment. These datasets function as ground truth, already labeled with attack and regular traffic. The selection of both datasets was based on their complementing features. The CSIC 2010 Dataset is well-known for including tens of thousands of actual HTTP traffic, encompassing both typical traffic and different kinds of cyberattacks. This makes it appropriate for assessing performance in real-world situations. In the meanwhile, the ECML/PKDD 2007 Challenge offers a dataset created especially to evaluate anomaly detection skills, which makes it a reliable standard for gauging how sensitive a model is to unexpected behavior. Partitioning this available data was part of the research process. The remaining 30% was kept as a holdout set to perform objective final performance assessment of the ML-RF model, while the remaining 70% was utilized to enrich the training data with other attack types.

The main goal of this study is to evaluate how well ML-RF detects cyberattacks. Automatic detection, system monitoring, and anomaly analysis are all part of control SI-4 (System and Communications Protections), which is the architecture outlined in NIST SP 800-53 Rev.5. The NIST SP 800-53 Rev.5-based methodology is used in this study to examine how well the two systems identify cyberattacks. The accuracy and efficacy of detection were tested utilizing assessment measures such the Confusion Matrix, Precision, Recall, False Positive Rate (FPR), and F-1 Score. Figure 1 depicts a sequence of steps that make up the research flow.

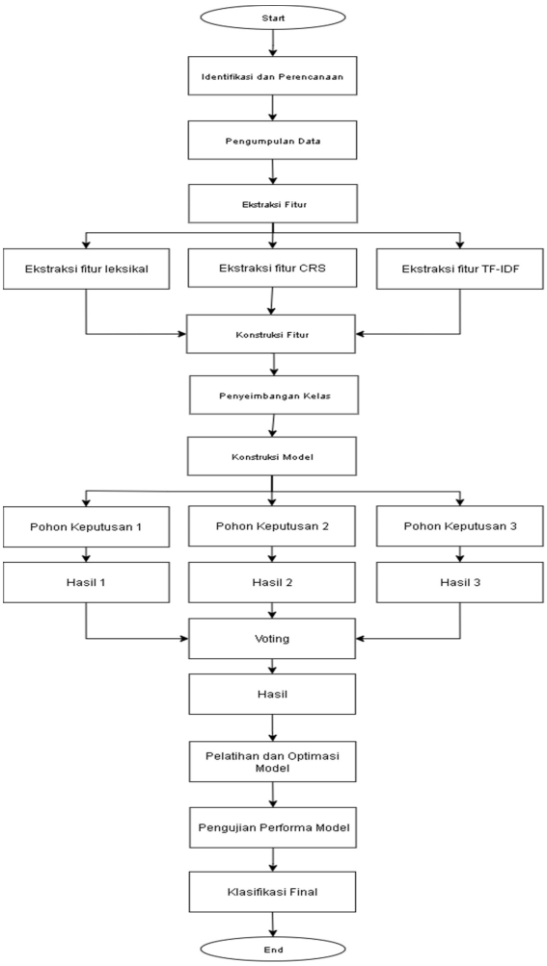

Figure 1. Methods of Research

3 RESULTS AND DISCUSSIONS

For universities like Jenderal Soedirman University (Unsoed), cyber security is essential to safeguarding the availability, confidentiality, and integrity of digital information assets. The Eldiru information system, which is safeguarded by the ModSecurity Web Application Firewall (WAF) using conventional OWASP Core Rule Set (CRS) rules, is one of the important assets that is the subject of this investigation. This setup exemplifies standard procedures used in the cybersecurity sector to counteract online threats.

However, a major operational problem was found during the practical testing of the conventional ModSecurity-CRS implementation in this study. The test results demonstrate that, in spite of their great accuracy, rule-based detection systems have a relatively poor recall rate (5.34%), as will be discussed in more depth in Sub-section 4.5. According to this, over 94% of real attacks go undetected, which is a catastrophic weakness that renders the system unreliable for complete security.

WAF is a passive system against most attacks because of this recall flaw. Its capacity to proactively detect assaults is somewhat restricted, despite its great blocking precision. This implies that the Eldiru system is still susceptible to a number of undiscovered attack methods that may be used against it without the defense system issuing any alerts.

The study developed a way to address these findings by incorporating machine learning to increase detection accuracy. In order to make more contextual classification judgments, the Random Forest algorithm is used in the hybrid detection module, ML-RF, which is the suggested method. In order to achieve a more balanced and useful security posture, the primary goal of this research is to develop, put into practice, and empirically assess a system that can significantly reduce false positives while retaining efficient detection capabilities against real attacks.

The methodological design was followed, and the data gathering procedure was completed satisfactorily. Eldiru Unsoed's internal server logs and common public datasets served as the two primary sources of the study data. Three primary data sets were generated by this method and utilized in the study: public dataset partitions for

testing and training data enrichment, and raw Nginx log samples that were then processed through simulations to yield audit.log data. The Nginx web server hosting the Eldiru system's access.log file served as the main source of data for model training. A random sample of 10 million entries was selected for the study from a data set of 77 million entries that were captured between December 2024 and January 2025. The purpose of this sampling is to lessen the impact of the computational resources' limits when processing the complete data set. A sample of the data set gathered during the data gathering phase is displayed in Table 1. A particular attribute of an HTTP transaction is represented by each column in Table 1. Table 2 offers thorough justifications for every attribute. The script does not use all 18 characteristics from access.log when simulating traffic replays. Rather, it uses the following essential properties to create a fresh HTTP request: Host, User-Agent, Referer, Client IP (included in the X-Forwarded-For header), and Request Line (which includes the method, URI, and HTTP version).

Table 1. Excerpt from the Eldiru dataset

Nomor	1	2	3	4	5
Cache Status	MISS	MISS	MISS	MISS	MISS
Upstream Length	276	276	276	564	564
Upstream Time	0.001	0.0.001	0.001	0.000	0.000
Upstream Status	403	403	403	200	200
Upstream Address	172.26.6.100:8000	172.26.6.100:8000	172.26.6.100:8000	172.26.6.100:8000	172.26.6.100:8000
Request Time	0.001	0.001	0.001	0.000	0.000
Server Name	eldiru.unsoed.ac.id	eldiru.unsoed.ac.id	eldiru.unsoed.ac.id	eldiru.unsoed.ac.id	eldiru.unsoed.ac.id
Host	eldiru.unsoed.ac.id	eldiru.unsoed.ac.id	eldiru.unsoed.ac.id	eldiru.unsoed.ac.id	eldiru.unsoed.ac.id
X-Forwarded-For	-	-	-		
User-Agent	Mozilla/5.0 AppleWebKit/537.36 (KHTML, like Gecko;	Mozilla/5.0 (X11; Linux x86_64)	Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)	Mozilla/5.0 (Linux; Android 14; V2327 Build/UP1A.231005.007; wv)	Mozilla/5.0 (Linux; Android 14; RMX3630 Build/UKQ1.230924.001; wv)
Referer	-	-	-	-	
Respons	276	276	276	564	564
Code	403	403	403	200	404
Request Line	GET /course/view.php?id=328 77 HTTP/1.1	GET /.env.development HTTP/1.1	GET /.env.old HTTP/1.1	POST webservice/rest/server.php?moo dlewsrestformat=json&wsfuncti on=tool_mobile_call_external_functions HTTP/1.1	POST /webservice/rest/server.php?moo dlewsrestformat=json&wsfuncti on=core_enrol_get_enrolled_users HTTP/1.1
Timestamp	19/Feb/2025:08:15:30 +0000	22/Jan/2025:14:30:15 +0000	05/Feb/2025:23:59:01 +0000	07/Jan/2025:11:11:11 +0000	28/Feb/2025:00:05:45 +0000
User Klien		-	-	-	-
Logname Klien	-	-	-	-	-
IP Klien	52.167.144.147	178.128.109.236	178.128.109.236	36.73.35.46	140.213.165.83

Table 2. Eldiru Dataset Attributes Explained

Column Name	Brief Explanation
IP Client	The IP address of the user's computer that sent the request.
Logname Client	The client log name from identd.
User Klient	The user name if the request uses basic HTTP authentication.
Timestamp	The time and date when the server finished processing the request.
Request Line	The complete request line from the client, containing the method, URI,
Status Code	and HTTP version.

Ukuran Respons	A three-digit code indicating the result of the request.
Referer	The size of the response body in bytes sent to the client.
User-Agent	The URL of the previous page where the client clicked a link to this page.
X-Forwarded-For	Information about the browser and operating system
Host	used by the client.
Server Name	The client's original IP address if the request passed through one or
Request Time	more proxies.
Upstream Address	The domain name requested by the client in the Host header.
Upstream Status	The name of the virtual server in Nginx that handled the request.
Upstream Time	The total time in seconds that the server took to process the request.
Upstream Length	The IP address and port of the backend server (e.g., application) that Nginx contacted.
Cache Status	The status code received by Nginx from the backend server.

3.1. Log Parsing and Labeling Based on Anomaly Scores

Parsing the combined training data from the audit.log replay results and public data is the first step in the procedure. A total of 528,622 items with ModSecurity warnings were successfully retrieved from the two log

```
================================================================================
MEMUAT SEMUA DATA LATIH
================================================================================
Ditemukan 2 file log: ['test_1.log', 'test_2.log']
Mem-parsing file log: test_1.log...
Selesai mem-parsing test_1.log. Ditemukan 311022 entri log dengan peringatan ModSecurity.
Mem-parsing file log: test_2.log...
Selesai mem-parsing test_2.log. Ditemukan 217600 entri log dengan peringatan ModSecurity.

Memuat dataset CSV: Data Latih CSV (70%) dari .\data_70.csv...
Selesai memuat Data Latih CSV (70%). Ditemukan 59469 baris.

Data latih berhasil digabungkan. Total baris: 588091
Distribusi label di data latih:
label
0    490406
1     97685
```

files. The final training dataset consisted of 588,091 transactions after this data was joined with 59,469 lines of data from the public dataset. Based on the CRS anomaly score, each transaction in this training data was then automatically labeled as either "Normal" (0) or "Anomaly" (1). This produced an initial class distribution of 490,406 "Normal" data and 97,685 "Anomaly" data. Figure 5 displays the results of this computerized parsing and labeling procedure that is based on anomaly scores.

Figure 2. Output Labeling

3.2. Extraction of Features

Extracting useful information from each request is the next step after data collection. The elbow method algorithm is used to automatically choose features in CRS rules in order to identify the most efficient and informative number of top rules. Extracting useful information from each request is the next step after data collection. The elbow method algorithm is used to automatically choose features in CRS rules in order to identify the most efficient and informative number of top rules. This function's execution results in a visual plot shown in Figure 3 and textual recommendations for the ideal number of rules shown in Figure 4. The three feature types—lexical, CRS rules, and N-gram TF-IDF—are retrieved and effectively merged into a single function after the ideal number of rules has been established. Figure 5 displays the output of the feature extraction process. Using RandomUnderSampler from the imblearn package, the under-sampling strategy was used to keep the model from getting biased towards the majority class (Normal). As seen in Figure 3, this method progressively lowers the proportion of samples from the majority class until a more balanced ratio with the minority class (Anomaly) is reached.

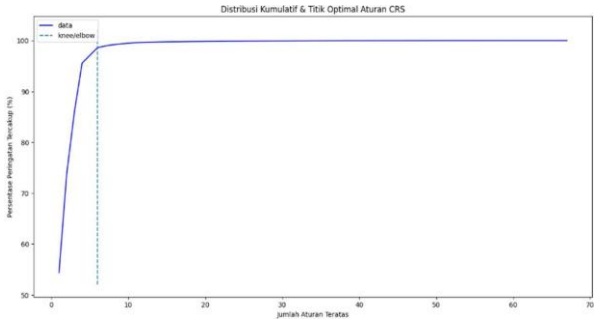

Figure 3. Optimal Visual Plot CRS Rules

```
================================================================================
MENGHITUNG JUMLAH ATURAN OPTIMAL SECARA OTOMATIS
================================================================================
☑ REKOMENDASI OTOMATIS: Gunakan 6 aturan teratas.
   -> Ini sudah mencakup 98.60% dari total semua peringatan.
--------------------------------------------------------------------------------
```

Figure 4. CRS Feature Selection Script Output

```
Memulai training model RandomForest...
Fitting 3 folds for each of 5 candidates, totalling 15 fits
[CV] END class_weight=balanced_subsample, max_depth=None, n_estimators=172; total time= 2.8min
[CV] END class_weight=balanced_subsample, max_depth=None, n_estimators=172; total time= 3.3min
[CV] END class_weight=balanced_subsample, max_depth=None, n_estimators=172; total time= 1.9min
[CV] END class_weight=balanced_subsample, max_depth=None, n_estimators=186; total time= 3.2min
[CV] END class_weight=balanced_subsample, max_depth=None, n_estimators=186; total time= 3.6min
[CV] END class_weight=balanced_subsample, max_depth=None, n_estimators=186; total time= 2.2min
[CV] END class_weight=balanced_subsample, max_depth=30, n_estimators=100; total time= 1.1min
[CV] END class_weight=balanced_subsample, max_depth=30, n_estimators=100; total time= 1.3min
[CV] END class_weight=balanced_subsample, max_depth=30, n_estimators=100; total time= 1.0min
[CV] END class_weight=balanced_subsample, max_depth=None, n_estimators=201; total time= 3.5min
[CV] END class_weight=balanced_subsample, max_depth=None, n_estimators=201; total time= 3.8min
[CV] END class_weight=balanced_subsample, max_depth=None, n_estimators=201; total time= 2.3min
[CV] END class_weight=balanced_subsample, max_depth=None, n_estimators=154; total time= 2.7min
[CV] END class_weight=balanced_subsample, max_depth=None, n_estimators=154; total time= 3.0min
[CV] END class_weight=balanced_subsample, max_depth=None, n_estimators=154; total time= 1.8min

Hyperparameter terbaik: {'class_weight': 'balanced_subsample', 'max_depth': None, 'n_estimators': 186}
```

Figure 5. Feature Extraction Output

```
================================================================================
METODE SAMPLING: Random Under-Sampling
Distribusi kelas SEBELUM under-sampling:
label
0    490406
1     97685

Distribusi kelas SETELAH under-sampling:
label
0    195370
1     97685
================================================================================
```

Figure 6. Random Sampling Output

3.3. Model Construction

The Scikit-learn package is used to build the model optimization procedure, which includes class balance and hyperparameter search. The TfidfVectorizer object and the learned RandomForestClassifier model (best_rf) are saved to a file using joblib as an essential last step in the implementation. This allows them to be reloaded without requiring a new training process. The result is displayed in Figure 7.

```
Memulai training model RandomForest...
Fitting 3 folds for each of 5 candidates, totalling 15 fits
[CV] END class_weight=balanced_subsample, max_depth=None, n_estimators=172; total time= 2.8min
[CV] END class_weight=balanced_subsample, max_depth=None, n_estimators=172; total time= 3.3min
[CV] END class_weight=balanced_subsample, max_depth=None, n_estimators=172; total time= 1.9min
[CV] END class_weight=balanced_subsample, max_depth=None, n_estimators=186; total time= 3.2min
[CV] END class_weight=balanced_subsample, max_depth=None, n_estimators=186; total time= 3.6min
[CV] END class_weight=balanced_subsample, max_depth=None, n_estimators=186; total time= 2.2min
[CV] END class_weight=balanced_subsample, max_depth=30, n_estimators=100; total time= 1.1min
[CV] END class_weight=balanced_subsample, max_depth=30, n_estimators=100; total time= 1.3min
[CV] END class_weight=balanced_subsample, max_depth=30, n_estimators=100; total time= 1.0min
[CV] END class_weight=balanced_subsample, max_depth=None, n_estimators=201; total time= 3.5min
[CV] END class_weight=balanced_subsample, max_depth=None, n_estimators=201; total time= 3.8min
[CV] END class_weight=balanced_subsample, max_depth=None, n_estimators=201; total time= 2.3min
[CV] END class_weight=balanced_subsample, max_depth=None, n_estimators=154; total time= 2.7min
[CV] END class_weight=balanced_subsample, max_depth=None, n_estimators=154; total time= 3.0min
[CV] END class_weight=balanced_subsample, max_depth=None, n_estimators=154; total time= 1.8min

Hyperparameter terbaik: {'class_weight': 'balanced_subsample', 'max_depth': None, 'n_estimators': 186}
```

Figure 7. Optimization of the Model and Storage Output

3.4. Evaluatino and Comparison

The suggested detection system's (ModSec-RF) performance was assessed and quantitatively contrasted with that of the conventional system (ModSecurity-CRS). To guarantee a fair and impartial comparison, the same test dataset (30% holdout set) was used to evaluate both algorithms. Figure 8 illustrates how the ModSec-CRS system's assessment process, which includes replay and output, is implemented. The script then creates a final performance report that includes a confusion matrix for the Standard CRS system by comparing these predicted labels with the original labels from the test data and the result are shown in Figure 9.

```
crs_label  Anomalous  Normal    All
Class
Anomalous        621   11019  11640
Valid             32   13816  13848
All              653   24835  25488

False Positive Rate (FPR): 0.23%
```

Figure 8. Test Data Replay Output

```
Memulai pemutaran ulang 25488 request ke http://localhost:80...
Proses pemutaran ulang selesai dalam 162.77 detik.
Hasil disimpan di: evaluation_result_FIXED.csv
```

Figure 9. CRS Testing Output

A confusion matrix, which thoroughly divides the classification findings into four quadrants—True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN)—is the primary depiction of these data. Tables 3 and 4 provide the confusion matrix data for the ML-RF system and the ModSecurity-CRS Standard system in consecutive order. Table 5 provides a more explicit contrast with the confusion matrix computations.

Table 3. ModSec-CRS Test Data Mapping

	Prediksi:Anomali	Prediksi: Valid	Total Aktual
Aktual: Anomali	621 (TP)	11019 (FN)	11640
Aktual: Valid	32 (FP)	13816 (TN)	13848
Total Prediksi	653	24835	25488

Table 4. ModSec-RF Test Data Mapping

	Prediksi: Anomali	Prediksi: Valid	Total Aktual
Aktual:Anomali	8.381 (TP)	3.259 (FN)	11.640
Aktual: Valid	831 (FP)	13.017 (TN)	13.848
Total Prediksi	9.212	16.276	25.488

Table 5. Confusion Matrix Comparison

Metrik	ModSecurity-CRS	ML-RF	Rumus
Precision	95.10%	91.00%	$\frac{TP}{FP+TP}$
Recall	5.34%	72.00%	$\frac{TP}{TP+FN}$
F1-Score	10.10%	80.00%	$2 \times \frac{Precision \times Recall}{Precision + Recall}$
Specificity (TNR)	99.77%	94.00%	$\frac{TN}{TN+FP}$
False Positive Rate	0.23%	6.00%	$\frac{FP}{FP+TN}$
Accuracy	56.64%	84.00%	$\frac{TP+TN}{Total\ Data}$

The stark differences in the two systems' performance characteristics are demonstrated by the empirical data in Table 5. The Standard CRS method has a very low False Positive Rate (0.23%) and extremely high accuracy (95.10%), but at the cost of a significantly lower recall (only 5.34%). This suggests that the CRS system is useless as a main protection system because, despite its infrequent blocking errors, it misses most attacks. On the other hand, the ML-RF system performs noticeably better in terms of balance and efficiency. This system has a recall value of 72.00%, which is greater than CRS, but having a little lower accuracy (91.00%). An F1-Score of 80.00% (as opposed to 10.10% for CRS) demonstrates this ideal balance between precision and recall, demonstrating that ML-RF is a far more dependable and efficient detection method overall.

From a security standpoint, the SHAP plot visualization validates that the model has picked up semantically meaningful patterns. High feature values (red color) continuously push predictions in the direction of anomalies (positive SHAP values), with lexical characteristics like special_chars_count and url_length occupying the top places. Strong signs were also found in the presence of particular N-grams, such as c%2 (URL representation for the % character), %2f (for /), and %22 (for "), which are often used URL encoding strategies in assaults. An interpretive study was conducted utilizing SHAP (SHapley Additive exPlanations) to comprehend the foundation of the ML-RF model's decision-making. Researchers may quantify each feature's contribution to the final prediction using this technique. In Figure 10, the analysis's findings are displayed as a SHAP summary plot.

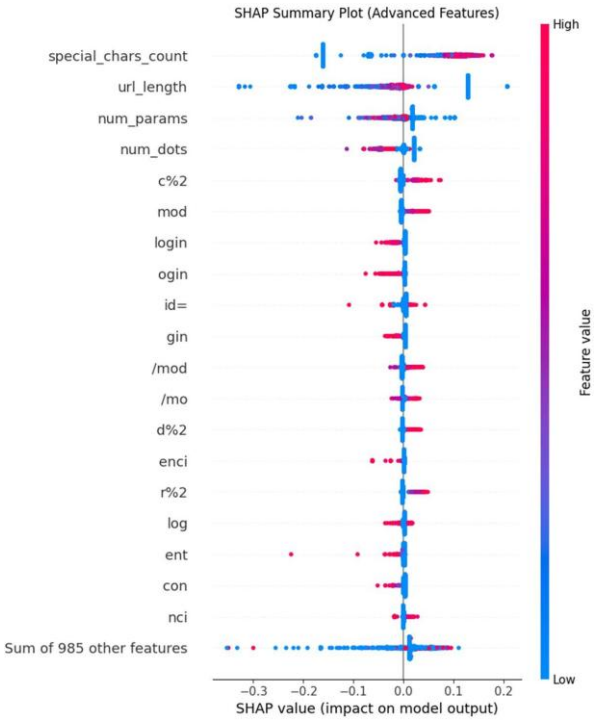

Figure 10. SHAP Plot Visualisation.

3.5. Extraction of Features

The last phase of this study was applying the ML-RF hybrid detection system to actual data after the model was verified and its performance assessed using publicly available test data in the preceding subchapter. The goal of this stage was to show how the system could recognize and categorize possible threats coming from actual traffic on the Eldiru Unsoed server that had no labels at first (unlabelled traffic). Approximately one million distinct transactions make up the dataset utilized in this implementation, which was produced by sampling 77 million access.log samples. This dataset offers a pertinent and realistic foundation for testing as it accurately depicts the system's actual operating circumstances. Figure 11 provides a description of the categorization findings.

By automatically scanning all CRS rule files and connecting each rule ID with an anomalous score determined by each rule's severity, this score dictionary is created. Following the completion of the classification procedure, a second script is executed to summarize the output data and carry out statistical analysis. Figure 11 displays the results of the statistical study. The results in Table 7 demonstrate that the algorithm may give classic attacks that are obviously harmful extremely high ratings. The CRS Score is the main factor in both situations, demonstrating its usefulness as a safeguard against known-pattern risks. In the meanwhile, Table 8 details instances when the ML Score plays a major role in identifying anomalies that nearly crossed the threshold, demonstrating the enhanced value of the machine learning model. Table 9 displays an aggregate analysis that maps all abnormalities found into distinct assault types.

```
--- Menganalisis File Hasil: hasil_analisis.csv ---

[1] Informasi Dasar & Cek Data Hilang
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 999888 entries, 0 to 999887
Data columns (total 5 columns):
 #   Column          Non-Null Count   Dtype
---  ------          --------------   -----
 0   request         999888 non-null  object
 1   crs_score       999888 non-null  int64
 2   ml_score        999888 non-null  float64
 3   hybrid_score    999888 non-null  float64
 4   classification  999888 non-null  object
dtypes: float64(2), int64(1), object(2)
memory usage: 38.1+ MB

[2] Distribusi Klasifikasi Final (Jumlah Anomali vs. Normal)
                Jumlah  Persentase (%)
classification
Normal          978458          97.86
Anomali          21430           2.14

[3] Spot-Check Data

--- 5 Contoh Deteksi Anomali (Kontribusi Utama dari ML) ---
Kasus di mana Skor CRS rendah (< 5), tapi Skor ML tinggi.
                  request  crs_score  ml_score  hybrid_score classification
31470  POST /cms HTTP/1.1          0      10.0          10.0        Anomali
33304  POST /cms HTTP/1.1          0      10.0          10.0        Anomali
45901  POST /cms HTTP/1.1          0      10.0          10.0        Anomali
58178  POST /cms HTTP/1.1          0      10.0          10.0        Anomali
58186  POST /cms HTTP/1.1          0      10.0          10.0        Anomali

--- 5 Contoh Deteksi Anomali (Kontribusi Utama dari CRS) ---
Kasus di mana Skor ML rendah (< 2.0), tapi Skor CRS tinggi.
                                                  request  crs_score  ml_score  hybrid_score classification
34363   GET /login.action?redirect:$%7B%23a%3D(new%20j...         25      1.72         26.72        Anomali
954579  GET /theme/yui_combo.php?3.18.1/calendar-base/...         25      0.70         25.70        Anomali
954871  GET /mod/forum/search.php?forumid=201&fromday=...         25      0.91         25.91        Anomali
954913  GET /lib/ajax/service-nologin.php?args=%5B%7B%...         25      0.48         25.48        Anomali
955093  GET /mod/forum/search.php?forumid=201&fromday=...         25      0.91         25.91        Anomali
```

Figure 11. Analysis Output for Classification Results

Table 6. Final Distribution

Klasifikasi	Jumlah	Persentase
Normal	978.456	97.86
Anomali	21.430	2.14
Total	999.888	100.00

Table 7. Sample Data 1

No	Request	Skor CRS	Skor ML	Skor Hybrid
1	GET /?order_id=%27%3E%22%3Csvg%2Fonload=confirm%28%27XSS%27%29%3E HTTP/1.1	85.0	7.53	92.53
2	GET /index.php?lang=../../../../../../../etc/passwd HTTP/1.1	48.0	8.76	56.76
3	GET /login.do?jvar_page_title=%3Cstyle%3E%3Cj%3Ajelly%20xmlns%3Aj%3D%22jelly%3Acore%22%3E%3C%2Fj%3Ajelly%3E%3C%2Fstyle%3E HTTP/1.1	40.0	8.55	48.55
4	GET /course/search.php?search=%27%3BSELECT%20PG_SLEEP(20)-- HTTP/1.1	40.0	6.56	46.56
5	GET /upgrade/detail.jsp?id=1%20UNION%20SELECT%20md5(1)%20from%20users HTTP/1.1	30.0	9.78	39.78

Table 8. Sample Data 2

No	Request	Skor CRS	Skor ML	Skor Hybrid
1	POST /xmlrpc.php HTTP/1.1	0	7.15	7.15
2	GET /?author=1 HTTP/1.1	0	7.05	7.05
3	GET /index.php?option=com_myblog&task=../../../etc/passwd	3.0	4.80	7.80
4	GET /wp-includes/wlwmanifest.xml HTTP/1.1	5.0	6.88	11.88
5	GET /config/laravel/.env.bak HTTP/1.1	5.0	5.50	10.50

Table 9. OWASP Categories

No	Kategori OWASP	Jumlah	Persentase
1	Anomaly_ML_Only	13.436	62.7%
2	A01_Broken_Access_Control	5.542	25.9%
3	A05_Security_Misconfiguration	1.561	7.3%
4	A03_Injection	891	4.1%

4 CONCLUSION

This study used a hybrid detection strategy with success. To create a precise and contextual Random Forest classification model, the procedure included lexical feature analysis, context from the OWASP Core Rule Set (CRS) rules, and N-gram representations. Additionally, SHAP interpretation study verified that the model correctly learns the inherent features of malicious data in addition to relying on CRS signals. The performance of the suggested machine learning-based system (ML-RF) is noticeably better than that of the conventional ModSecurity implementation with CRS. The F1-Score increased from 10.10% to 80.00% and the recall measure increased from 5.34% to 72.00%, demonstrating this superiority. These outcomes demonstrate that the machine learning method may create a detection system that is more effective and balanced overall. Applying this analysis to actual data yielded further results. The model's capacity to recognize hazards that elude rule-based detection was demonstrated by the fact that 62.7% of the identified anomalies were categorized as Anomaly_ML_Only. However, this discovery also draws attention to the automated mapping process's shortcomings and the possibility of unclear outcomes. This high number could also suggest that the model is very sensitive to regular traffic patterns with huge volumes that are not well-represented in the training set, which could lead to false positives. The creation of a test sample dataset from server access logs and categorization using other machine learning techniques are suggestions for more study.

REFERENCES

[1] G. Tsochev, R. Trifonov, O. Nakov, S. Manolov, and G. Pavlova, "Cyber security: Threats and Challenges," *IEEE Explore*, 2020.

[2] T. Y. Kim and S. B. Cho, "Optimizing CNN-LSTM neural networks with PSO for anomalous query access control," *Neurocomputing*, vol. 456, pp. 666–677, Oct. 2021, doi: 10.1016/j.neucom.2020.07.154.

[3] N. Gupta, V. Jindal, and P. Bedi, "LIO-IDS: Handling class imbalance using LSTM and improved one-vs-one technique in intrusion detection system," *Computer Networks*, vol. 192, Jun. 2021, doi: 10.1016/j.comnet.2021.108076.

[4] X. Song, C. Chen, B. Cui, and J. Fu, "Malicious javascript detection based on bidirectional LSTM model," *Applied Sciences (Switzerland)*, vol. 10, no. 10, May 2020, doi: 10.3390/app10103440.

[5] H. S. Obaid and E. H. Abeed, "Abeed,-DoS and DDoS Attacks at OSI Layers," *International Journal of Multidisciplinary Research and Publications Hadeel S. Obaid and Esamaddin H*, vol. 2, no. 8, pp. 1–9, 2020, doi: 10.5281/zenodo.3610833.

[6] A. A. Mughal, "Cyber Attacks on OSI Layers: Understanding the Threat Landscape," 2020. [Online]. Available: https://orcid.org/0009-0006-8460-8006

[7] G. C. Amaizu, C. I. Nwakanma, S. Bhardwaj, J. M. Lee, and D. S. Kim, "Composite and efficient DDoS attack detection framework for B5G networks," *Computer Networks*, vol. 188, Apr. 2021, doi: 10.1016/j.comnet.2021.107871.

[8] C. Luo, Z. Tan, G. Min, J. Gan, W. Shi, and Z. Tian, "A Novel Web Attack Detection System for Internet of Things via Ensemble Classification," *IEEE Trans Industr Inform*, vol. 17, no. 8, pp. 5810–5818, Aug. 2021, doi: 10.1109/TII.2020.3038761.

[9] F. Yasin, Abdul Fadlil, and Rusydi Umar, "Identifikasi Bukti Forensik Jaringan Virtual Router Menggunakan Metode NIST," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 5, no. 1, pp. 91–98, Feb. 2021, doi: 10.29207/resti.v5i1.2784.

[10] M. S. Rafsanjani, V. Suryani, and R. R. Pahlevi, "Deteksi Serangan Botnet Pada Jaringan Internet of Things Menggunakan Algoritma Random Forest (RF)."

[11] G. Xu *et al.*, "JSCSP: A Novel Policy-Based XSS Defense Mechanism for Browsers," *IEEE Trans Dependable Secure Comput*, vol. 19, no. 2, pp. 862–878, 2022, doi: 10.1109/TDSC.2020.3009472.

[12] K. Vijayalakshmi and E. Syed Mohamed, "Case Study: Extenuation of XSS Attacks through Various Detecting and Defending Techniques," *Journal of Applied Security Research*, vol. 16, no. 1, pp. 91–126, 2021, doi: 10.1080/19361610.2020.1735283.

[13] A. Fidalgo, I. Medeiros, P. Antunes, and N. Neves, "Towards a Deep Learning Model for Vulnerability Detection on Web Application Variants," in *Proceedings - 2020 IEEE 13th International Conference on Software Testing, Verification and Validation Workshops, ICSTW 2020*, Institute of Electrical and Electronics Engineers Inc., Oct. 2020, pp. 465–476. doi: 10.1109/ICSTW50294.2020.00083.

[14] D. Faroek, Rusydi Umar, and Imam Riadi, "Classification Based on Machine Learning Methods for Identification of Image Matching Achievements," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 6, no. 2, pp. 198–206, Apr. 2022, doi: 10.29207/resti.v6i2.3826.

[15] S. A. Reddy and B. Rudra, "Evaluation of Recurrent Neural Networks for Detecting Injections in API Requests," in *2021 IEEE 11th Annual Computing and Communication Workshop and Conference, CCWC 2021*, Institute of Electrical and Electronics Engineers Inc., Jan. 2021, pp. 936–941. doi: 10.1109/CCWC51732.2021.9376034.

[16] Dwi Kurnia Wibowo, Ahmad Luthfi, Yudi Prayudi, Erika Ramadhani, and Muhamad Maulana, "Faux Insider Hazard Investigation on Non-Public Cloud Computing by Using ADAM's Technique," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 6, no. 6, pp. 1028–1036, Dec. 2022, doi: 10.29207/resti.v6i6.4714.

[17] A. Mihoub, O. Ben Fredj, O. Cheikhrouhou, A. Derhab, and M. Krichen, "Denial of service attack detection and mitigation for internet of things using looking-back-enabled machine learning techniques," *Computers and Electrical Engineering*, vol. 98, Mar. 2022, doi: 10.1016/j.compeleceng.2022.107716.

[18] C. S. Nwosu, S. Dev, P. Bhardwaj, B. Veeravalli, and D. John, "Predicting Stroke from Electronic Health Records," in *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, IEEE, 2019, pp. 5704–5707. doi: 10.1109/EMBC.2019.8857234.

[19] Z. shi Gao, Y. Su, Y. Ding, Y. dong Liu, X. an Wang, and J. wei Shen, "Key Technologies of Anomaly Detection Using PCA-LSTM," in *Advances in Intelligent Systems and Computing*, Springer Verlag, 2020, pp. 246–254. doi: 10.1007/978-3-030-22263-5_24.

[20] P. Roy, R. Kumar, and P. Rani, "SQL Injection Attack Detection by Machine Learning Classifier," in *Proceedings - International Conference on Applied Artificial Intelligence and Computing, ICAAIC 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 394–400. doi: 10.1109/ICAAIC53929.2022.9792964.

[21] M. S. Elsayed, N. A. Le-Khac, S. Dev, and A. D. Jurcut, "DDoSNet: A Deep-Learning Model for Detecting Network Attacks," in *Proceedings - 21st IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM 2020*, Institute of Electrical and Electronics Engineers Inc., Aug. 2020, pp. 391–396. doi: 10.1109/WoWMoM49955.2020.00072.

[22] D. Chen, Q. Yan, C. Wu, and J. Zhao, "SQL Injection Attack Detection and Prevention Techniques Using Deep Learning," in *Journal of Physics: Conference Series*, IOP Publishing Ltd, Feb. 2021. doi: 10.1088/1742-6596/1757/1/012055.

[23] X. Du *et al.*, "SoK: Exploring the state of the art and the future potential of artificial intelligence in digital forensic investigation," *ACM International Conference Proceeding Series*, 2020, doi: 10.1145/3407023.3407068.
[24] F. Kovacs and G. Gcfa, "Windows 10 as a Forensic Platform," 2021.
[25] V. R. Kebande, R. A. Ikuesan, N. M. Karie, S. Alawadi, K. K. R. Choo, and A. Al-Dhaqm, "Quantifying the need for supervised machine learning in conducting live forensic analysis of emergent configurations (ECO) in IoT environments," *Forensic Science International: Reports*, vol. 2, 2020, doi: 10.1016/j.fsir.2020.100122.
[26] L. Tageldin and H. Venter, "Machine-Learning Forensics: State of the Art in the Use of Machine-Learning Techniques for Digital Forensic Investigations within Smart Environments," *Applied Sciences (Switzerland)*, vol. 13, no. 18, p. 10169, 2023, doi: 10.3390/app131810169.
[27] L. F. Sikos, "Packet analysis for network forensics: A comprehensive survey," Mar. 01, 2020, *Elsevier Ltd*. doi: 10.1016/j.fsidi.2019.200892.