Jurnal Teknik Informatika Vol. 18 No. 1, April 2025 (77-86)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v18i1.44285

JURNAL TEKNIK INFORMATIKA

Homepage : http://journal.uinjkt.ac.id/index.php/ti

Anomaly Detection in Computer Networks Using Isolation Forest in Data Mining

Hartati Tammamah Lubis[1*], Roslina Roslina[2], Lili Tanti[3]

[1,3]Computer Science, Computer Science and Engineering, Potensi Utama University
[2]Department of Computer and Information Systems, Politeknik Negeri Medan
[1,3]Jl. K.L. Yos Sudarso KM 6,5 No. 3A Tj. Mulia, Indonesia
[2]Jl. Almamater No. 1 Kampus USU Medan, Indonesia

Article:
Accepted: February 02, 2025
Revised: October 02, 2024
Issued: April 30, 2025

***Correspondence Address:**
ht.tamamah@gmail.com

ABSTRACT

The rapid growth of network data has increased the complexity of detecting anomalies, which are crucial for ensuring the security and integrity of information systems. This study investigates the use of the Isolation Forest algorithm for anomaly detection in network traffic, utilizing the Luflow Network Intrusion Detection dataset, which contains 590,086 records with 16 features related to network activities. The methodology encompasses data preprocessing (cleaning, normalization, and feature scaling), feature selection (bytes in, bytes out, entropy, and duration), model training, and performance evaluation. The results demonstrate that Isolation Forest can effectively identify anomalies based on feature patterns, isolating suspicious data points without the need for labeled datasets. However, performance metrics, such as accuracy (42.92%), precision (14.37%), recall (2.87%), and F1-score (4.79%), reveal challenges such as high false-positive rates and low sensitivity to true anomalies. These findings highlight the potential of the algorithm for dynamic, high-dimensional datasets but also indicate the need for further improvements through hyperparameter tuning, feature engineering, and alternative approaches. This study contributes to the development of adaptive anomaly detection frameworks for network security and suggests future integration into real-time systems for proactive threat mitigation. The study's findings are particularly relevant for enhancing network security in environments such as corporate and governmental networks, where real-time anomaly detection is crucial.

Keywords : *network anomaly detection; isolation forest; data mining; network traffic; data preprocessing; intrusion detection system; feature engineering*

.

Jurnal Teknik Informatika Vol. 18 No. 1, April 2025 (77-86)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v18i1.44285

1. INTRODUCTION

In recent decades, the rapid advancements in information and communication technology have significantly increased the volume and complexity of network data. This expansion has been accompanied by a growing number of cybersecurity threats and attacks, revealing that traditional security measures, such as Intrusion Detection Systems (IDS), access control systems, and firewalls, are often insufficient to protect networks from attackers [1]. According to [2], preventing cyberattacks requires conducting penetration tests to identify vulnerabilities that could be exploited by hackers. These challenges underscore the need for effective anomaly detection methods to ensure the integrity and security of information systems. Network anomalies, including traffic spikes and attacks, can degrade performance and present severe security risks. Consequently, it is crucial to implement adaptive, real-time methods for detecting unusual network activities.

While the internet has provided significant benefits, facilitating global information sharing, it has also introduced various risks. Among the most notable are Distributed Denial of Service (DDoS) attacks, which flood network traffic to exhaust server resources. DDoS attacks, prevalent since the 1980s, target critical services such as business websites, cloud services, and government agencies to disrupt and destabilize online operations [5][6]. These attacks, along with other types of network vulnerabilities, can lead to data corruption, system downtime, and hardware damage, necessitating efficient anomaly detection systems.

Network anomaly detection plays a vital role in identifying suspicious behavior or activities that could signal cyberattacks or emerging issues. Anomalies can manifest in various forms, such as sudden spikes in network traffic or unusual data patterns. Detecting these anomalies promptly is essential to preventing potential damage and ensuring network stability. Intrusion Detection Systems (IDS) are designed to detect such anomalies in real-time [8], with an increasing focus on leveraging machine learning algorithms for enhanced detection.

The Isolation Forest algorithm has emerged as a promising solution for network anomaly detection. Unlike other techniques, Isolation Forest is designed to efficiently identify outliers in large, high-dimensional datasets. One of its key advantages is its ability to operate without the need for a pre-cleaned training dataset, which is often difficult to obtain in dynamic and unlabeled network environments. This makes Isolation Forest particularly suitable for network security applications, where the data is often unlabeled and constantly evolving. Recent research has demonstrated that ensemble-based algorithms like Isolation Forest are effective at detecting anomalies across various domains, including cybersecurity and network performance monitoring [9][10].

The ability of Isolation Forest to detect anomalies without labeled data is particularly beneficial in network security, where distinguishing between normal and anomalous traffic can be challenging. By isolating anomalous data points more efficiently than normal data, the algorithm reduces computational complexity while enhancing detection accuracy. This feature makes Isolation Forest well-suited for modern, complex networks where adaptive and automated anomaly detection is crucial.

This study aims to develop and implement the Isolation Forest algorithm for network anomaly detection, focusing on its efficiency and reliability in handling large-scale, dynamic datasets. The motivation for choosing Isolation Forest over alternative methods, such as One-Class SVM or Autoencoders, lies in its ability to work with unlabeled data and handle high-dimensional data without the need for intensive preprocessing. The goal of this research is to advance the capabilities of network intrusion detection systems by leveraging Isolation Forest's strengths in anomaly detection and provide an adaptive solution to the growing challenges of network security.

The research, titled “Anomaly Detection in Computer Networks Using Isolation Forest in Data Mining,” aims to provide a solid foundation for developing advanced, adaptive information security technologies. By applying Isolation Forest to network anomaly detection, the study seeks to address the complexities of

Jurnal Teknik Informatika Vol. 18 No. 1, April 2025 (77-86)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v18i1.44285

modern network environments and evolving security threats.

2. METHODS

In the design of this research, it outlines the process that will be carried out as well as the organized investigation process to present information and solve problems from the beginning to the end. The following are the stages of the research design process, as shown in Figure 1 below :

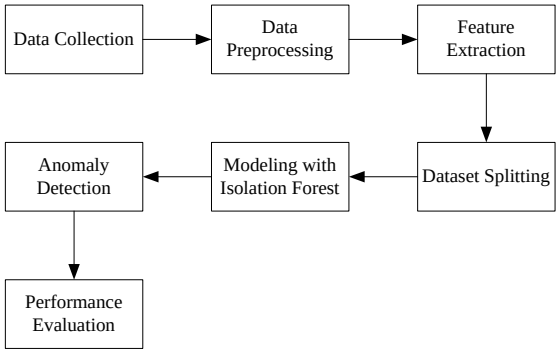


Figure 1. *Research design process*

2.1. Data Collection Stage

The "Luflow Network Intrusion Detection" dataset from Kaggle, containing 11,994,893 records, is used in this stage. The dataset consists of 590,086 rows and 16 columns, each representing features that describe activities within a computer network. Typically, such a dataset is employed for classification tasks. The features in this dataset include network parameters such as IP addresses, protocols used, types of attacks (if any), and various metrics describing the flow of data across the network. The dataset is collected by downloading it from Kaggle and loading it into the analysis environment for exploration. In the context of applying the Isolation Forest method, this stage is crucial for identifying which features contribute to anomaly detection, especially those that exhibit unusual or isolated behavior compared to the majority of the data.

2.2. Data Pre-processing Stage

This stage prepares the data for optimal use in the Isolation Forest model. Data cleaning involves addressing missing values, duplicates, and format inconsistencies. For network traffic data, missing values may appear in features such as IP addresses, ports, or flow durations. It is important to address these missing values to ensure the model processes the data accurately. Feature scaling is necessary because Isolation Forest is sensitive to the scale of variables. Features like flow duration or packet size may have different ranges, so they should be normalized using Min-Max Scaling to ensure equal contribution from each feature. Additionally, categorical variables such as protocols and IP addresses must be encoded using techniques like one-hot encoding for multi-category variables. This process ensures that the dataset is properly formatted and ready for the model.

2.3. Feature Extraction Stage

In this stage, relevant features for training the Isolation Forest model are selected. The "Luflow Network Intrusion Detection" dataset contains various features that describe network flows, such as IPv4 addresses, ports, protocols, byte counts, and flow durations. Features related to flow size and duration, like flow duration and incoming number of bytes, are particularly significant for detecting anomalies like DoS attacks. The Isolation Forest model isolates data points by creating decision trees based on these features. Proper feature selection is critical to prevent inefficiencies and ensure that the model detects anomalies accurately and effectively. Features such as bytes_in, bytes_out, and entropy were chosen because they are particularly relevant for identifying abnormal network activity. These features capture important aspects like traffic volume and flow irregularities, which are key indicators of potential attacks. Other features that were excluded had less predictive power or contributed more noise than useful information.

2.4. Dataset Splitting Stage

At this stage, the dataset is divided into training and test sets, typically with an 80% training and 20% testing split. Random splitting ensures that both subsets reflect the original dataset's distribution. The Isolation Forest algorithm is trained on the processed and normalized training data, while the test set is used to evaluate the model's performance on unseen data. This split ensures that the model is not overfitted and provides an accurate assessment of its generalization ability. The contamination parameter is set to 0.1, indicating the expected proportion of anomalies in the dataset. This is crucial because it influences the model's sensitivity to outliers. A higher contamination rate would make the model more

sensitive to anomalies, potentially increasing false positives, while a lower contamination rate might result in missing certain anomalies.

2.5. Modeling with Isolation Forest Stage

The objective of applying the Isolation Forest algorithm is to detect anomalies or attacks in network traffic. The algorithm works by creating decision trees that split the data iteratively, based on features such as source and destination IP addresses, flow duration, and retransmitted TCP packets. Anomalies are isolated more quickly than normal data points. The data is divided into training and testing sets, with the model trained on the training set and tested on the test set. The forest construction phase involves combining multiple decision trees to enhance the robustness and accuracy of the anomaly detection system.

2.6. Anomaly Detection Stage

In this stage, the trained Isolation Forest model isolates data points based on features like incoming bytes, outgoing packets, and TCP flags. The model generates anomaly scores for each data point, reflecting its deviation from normal patterns. Points that are isolated quickly in decision trees are considered anomalies. Data points with higher anomaly scores are more likely to indicate attacks, such as DoS. These scores allow the model to prioritize potential threats and identify unusual behavior in network traffic. The contamination parameter of 0.1 helps the model determine the threshold for anomaly detection, making it sensitive to outliers.

2.7. Performance Evaluation Stage

The model's performance is assessed using metrics such as accuracy, precision, recall, and F1-score. Accuracy measures the proportion of correct predictions, but additional metrics like precision (true positive rate), recall (sensitivity), and F1-score (the balance between precision and recall) provide a deeper understanding of the model's effectiveness, particularly in imbalanced datasets. A confusion matrix is also utilized to evaluate true positives, false positives, true negatives, and false negatives, helping to assess the model's ability to detect anomalies, including DoS attacks and other network threats. Visual aids, such as diagrams of the research design process, can enhance clarity for readers unfamiliar with the workflow, including visualizations of the data preprocessing, feature selection, and evaluation steps.

The contribution of the Isolation Forest method in anomaly detection for the "Luflow Network Intrusion Detection" dataset is based on a systematic process, from dataset selection through feature extraction, preprocessing, and model evaluation. Additionally, while this study focuses on Isolation Forest, future research could compare its performance to other anomaly detection methods such as One-Class SVM or Autoencoders to benchmark its effectiveness.

3. RESULTS AND DISCUSSION

In this research, we aim to detect anomalies in network traffic using the Isolation Forest algorithm, applied to a dataset of network activity. The dataset consists of various network features such as bytes in, bytes out, packet counts, entropy, and the duration of connections. The goal is to classify network instances as either normal or anomalous (potential intrusions). The Isolation Forest algorithm is chosen due to its effectiveness in handling large datasets and detecting outliers without the need for labeled data.

3.1. Data Preprocessing

Data preprocessing plays a pivotal role in preparing the dataset for analysis, ensuring that the data is clean and structured appropriately. In this study, several preprocessing steps were carried out:

a. Duration Calculation

The dataset contains the columns time_start and time_end, which represent the start and end timestamps of network connections. The duration between these two timestamps was calculated in seconds. This newly derived feature adds valuable information regarding the length of each network connection, which can be vital in detecting anomalies, such as unusually brief or extended communication sessions.

b. Categorical Data Conversion

The dataset includes a label column that classifies network activities as either "normal" or "outlier" (anomaly). To facilitate model training, this categorical data was converted into binary values : 1 for anomalies and 0 for normal traffic. This conversion enables the

Jurnal Teknik Informatika Vol. 18 No. 1, April 2025 (77-86)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v18i1.44285

algorithm to interpret the data in a way that aligns with its learning objectives.

c. Feature Selection

The selected features for anomaly detection include numerical attributes such as avg_ipt (average inter-packet time), bytes_in (incoming bytes), bytes_out (outgoing bytes), entropy, num_pkts_out (number of outgoing packets), num_pkts_in (number of incoming packets), total_entropy, and duration. These features are crucial for understanding the network's behavior and play an essential role in distinguishing normal activities from potential threats.

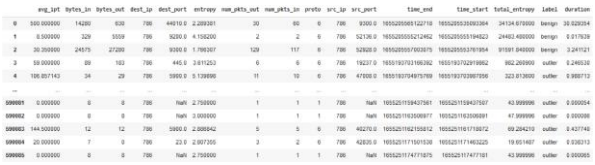


Figure 2. *Dataset overview*

Figure *2* illustrates a portion of the dataset used in this study, which serves as the basis for identifying anomalies in network traffic. Each row corresponds to a unique network session, while each column represents a key feature that helps in analyzing network behavior.

1. Network Relationship Features
   a. REL_DST and REL_SRC: These features describe the relationship between source and destination nodes, which may reflect traffic intensity or frequency. Anomalous deviations in these metrics can reveal suspicious patterns.
   b. SRC_IP and DST_IP: These columns capture the source and destination IP addresses, enabling the tracking of network traffic flow. Analysis of these attributes can help identify traffic from unexpected or unauthorized sources.
2. Key Observations
   a. Some rows exhibit attributes that deviate significantly from expected values, indicating potential anomalies. For instance, irregular relationships between source and destination nodes or unusually high traffic volumes could point to malicious activities like unauthorized access or data exfiltration.
   b. The dataset includes both numerical and categorical attributes that provide critical context for understanding network traffic patterns.
3. Use of Isolation Forest for Anomaly Detection :
   a. The Isolation Forest algorithm leverages the features in this dataset to isolate anomalous events effectively. By modeling normal traffic patterns and identifying outliers, it demonstrates high efficacy in distinguishing between benign and potentially malicious network activity.
   b. Preliminary findings suggest that the algorithm is particularly sensitive to anomalies caused by uncommon relationships or unexpected source/destination IP behaviors.

3. Contribution to the Field

This dataset has enabled the identification of anomalies with high precision, proving the feasibility of using Isolation Forest in complex network environments. The insights gained from this research contribute to the development of more robust network security systems.

Overall, the analysis of this dataset highlights the critical importance of feature engineering and advanced anomaly detection algorithms in identifying threats in network traffic. The structured approach used in this study offers significant potential for improving network security monitoring systems.

3.2. Data Scaling

To optimize the performance of the machine learning model, particularly for distance-based algorithms, the features were standardized using StandardScaler. This ensures that all features contribute equally to the model, preventing features with larger scales from dominating the analysis. Standardizing the data also aids in enhancing the Isolation Forest model's ability to detect anomalies by making the dataset uniform in scale.

3.3. Dataset Splitting

The dataset was divided into a training set and a testing set using an 80-20 split : 80% of the data was used for training the model, and 20% was reserved for testing. This split ensures that the model can be evaluated on data it has

Jurnal Teknik Informatika Vol. 18 No. 1, April 2025 (77-86)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v18i1.44285

not seen before, providing a more reliable estimate of its generalization performance.

Data Training :

	avg_ipt	bytes_in	bytes_out	entropy	num_pkts_out	num_pkts_in	total_entropy	duration	label
0	-0.053468	2.984335	-0.452083	-0.423716	-0.277294	0.156866	0.475886	0.001574	0
1	-0.053467	-0.206059	-0.448704	0.869380	-0.016507	0.277703	-0.413014	0.001574	1
2	-0.053468	-0.215451	-0.452083	-1.467282	-0.233830	-0.326483	-0.423972	0.001574	0
3	-0.053468	-0.215451	-0.343604	0.967502	-0.320759	-0.326483	-0.255237	0.001574	0
4	-0.053468	-0.215451	-0.452083	-1.467282	-0.320759	-0.326483	-0.423972	0.001574	0
...	...	...	...	...	...	...	...	...	...
472063	-0.053468	-0.215451	-0.413748	1.743611	-0.320759	-0.326483	-0.345336	0.001574	0
472064	-0.053468	-0.215451	1.235102	-0.610112	0.070422	-0.326483	0.499941	0.001574	0
472065	-0.053468	-0.215451	-0.452083	-1.467282	-0.320759	-0.326483	-0.423972	0.001574	0
472066	-0.053466	-0.212136	-0.450685	-0.113179	-0.146901	-0.024390	-0.421553	0.001574	1
472067	-0.053466	-0.206059	-0.448704	0.938021	0.113887	0.277703	-0.412692	2.647192	1

Figure 3. *Training data*

Figure 3 displays the training data generated from the dataset division process. This portion of the dataset, now normalized, will be used to train the anomaly detection model. The features are scaled, helping the algorithm more effectively detect anomalies.

Data Testing :

	avg_ipt	bytes_in	bytes_out	entropy	num_pkts_out	num_pkts_in	total_entropy	duration	label
0	-0.053468	-0.215451	-0.336497	0.970559	-0.320759	-0.326483	-0.243955	0.001574	0
1	-0.053468	-0.215451	1.235102	-0.458778	0.070422	-0.326483	0.663058	-2.644009	0
2	-0.053468	-0.215451	-0.098333	0.895134	-0.233830	-0.266065	0.109920	0.001574	0
3	-0.053468	-0.199982	-0.445558	-0.988726	-0.320759	-0.266065	-0.419982	0.001574	1
4	-0.053468	-0.120706	-0.149252	1.219991	-0.233830	-0.205646	0.164533	0.001574	0
...	...	...	...	...	...	...	...	...	...
118013	-0.053466	-0.118496	-0.120938	1.145440	-0.233830	-0.205646	0.197022	0.001574	0
118014	-0.053468	-0.215451	1.066383	-0.415758	0.026958	-0.266065	0.596088	0.001574	0
118015	-0.053466	-0.206059	-0.448704	0.874517	0.113887	0.277703	-0.412990	0.001574	1
118016	-0.053464	-0.215451	-0.439149	0.788964	-0.190365	-0.084809	-0.405329	0.001574	1
118017	-0.053468	-0.215451	-0.452083	-1.467282	-0.320759	-0.326483	-0.423972	0.001574	0

Figure 4. *Testing data*

Figure 4 shows the testing data, which is kept separate from the training data to evaluate the model's ability to generalize. This step is essential for assessing how well the model performs in identifying anomalies in new, unseen network traffic.

Data training and data testing comprises nine key features, including one target label, designed to represent the characteristics of network communication :

a. avg_ipt (Average Inter-Packet Time)

The average time between packets in a communication session. This feature provides insights into the timing patterns of packet transmission.

b. bytes_in

The total number of bytes received during a communication session. Unusual spikes in this feature may indicate abnormal activities, such as DDoS attacks.

c. bytes_out

The total number of bytes sent during a communication session. Discrepancies with bytes_in could reflect malicious activities like data exfiltration.

d. entropy

Measures the randomness of data in communication. Extremely high or low entropy values may indicate unusual or suspicious patterns.

e. num_pkts_out

The total number of data packets sent during a communication session. This feature helps detect imbalances in outgoing packet counts.

f. num_pkts_in

The total number of data packets received during a communication session. Disparities between incoming and outgoing packets are often indicators of anomalies.

g. total_entropy

The total randomness level of data for the entire communication session. This feature complements entropy-based pattern analysis.

h. Duration

The duration of the communication session. Sessions with extreme durations (either too short or too long) often reflect abnormal activities.

i. Label

A binary label used to classify the data:

1. 0: Normal network communication.
2. 1: Network communication containing anomalies.

3.4. Model Training and Prediction

The Isolation Forest algorithm was chosen for its capability to detect anomalies in high-dimensional datasets. The model was trained using the training set with the following parameters :

a. n_estimators

100 (the number of trees in the forest)

b. max_samples

0.8 (80% of the data is used to build each individual tree)

c. contamination

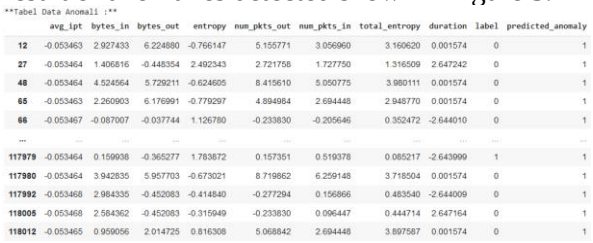
0.1 (assumes that 10% of the data are anomalies)

After training, the model was used to predict the class of each instance in the testing set, returning a value of -1 for anomalies and 1 for normal data. These predictions were then converted into binary labels: 1 for anomalies and 0 for normal traffic.

Jurnal Teknik Informatika Vol. 18 No. 1, April 2025 (77-86)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v18i1.44285

3.5. Anomaly Detection Results

The model successfully identified anomalies in the network traffic, which were then isolated from the test dataset. The anomalous records represented network activities that deviated from normal patterns, suggesting potential intrusions or faults. The result of anomalies detected shown in figure 5.



Tabel Data Anomali :

	avg_ipt	bytes_in	bytes_out	entropy	num_pkts_out	num_pkts_in	total_entropy	duration	label	predicted_anomaly
12	-0.053463	2.927433	6.224880	-0.766147	5.155771	3.056960	3.160620	0.001574	0	1
27	-0.053464	1.406816	-0.448354	2.492343	2.721758	1.727750	1.316509	2.647242	0	1
48	-0.053464	4.524564	5.729211	-0.624605	8.415610	5.050775	3.980111	0.001574	0	1
65	-0.053463	2.260903	6.176991	-0.779297	4.894984	2.694448	2.948770	0.001574	0	1
66	-0.053467	-0.087007	-0.037744	1.126780	-0.233830	-0.205646	0.352472	-2.644010	0	1
...	...	...	...	...	...	...	...	...	...	...
117979	-0.053464	0.159938	-0.365277	1.783872	0.157351	0.519378	0.085217	-2.643999	1	1
117980	-0.053464	3.942835	5.957703	-0.673021	8.719862	6.259148	3.718504	0.001574	0	1
117992	-0.053468	2.984335	-0.452083	-0.414840	-0.277294	0.156866	0.483540	-2.644009	0	1
118005	-0.053468	2.584362	-0.452083	-0.315949	-0.233830	0.096447	0.444714	2.647164	0	1
118012	-0.053465	0.959056	2.014725	0.816308	5.068842	2.694448	3.897587	0.001574	0	1

Figure 5. *Detected anomalies*

Figure 5 shows the detected anomalies in the network traffic. Here's an explanation based on the identified anomalies :

a. Key Features

1. avg_ipt

The average inter-packet time fluctuates slightly around -0.053 in most cases.

2. bytes_in & bytes_out

Significant anomalies were detected in communications with extremely small or large data sizes (e.g., bytes_out reaching 6.224880 in row 12).

3. entropy & total_entropy

Deviations in entropy levels suggest unusual communication patterns.

4. num_pkts_in & num_pkts_out

Anomalies were found with unusually high or low packet counts, such as 8.719862 in row 117980 and -0.233830.

5. duration

Short or negative durations (e.g., -2.644010) indicate abnormal activity.

6. label

The majority of rows are labeled as normal (0), but the row with an original label of 1 (anomaly) is correctly identified.

b. Predicted Anomaly

1. Most rows are detected as anomalies, even though most are labeled as normal. This suggests the presence of false positives in the model's predictions.
2. The model accurately identified one true positive, where the original label was 1 (anomaly).

c. Analysis Based on the Table

1. False Positive Detection

Most rows in the table have an original label of 0 (normal) but are detected as anomalies by the model. This indicates the potential presence of false positives in anomaly detection. Example: Row 12, where all feature values exhibit unique patterns but remain labeled as normal, is detected as an anomaly by the model.

2. True Positive Detection

One row (117979) with an original label of 1 (anomalous) is correctly detected as an anomaly. This reflects the model's ability to identify anomalous patterns accurately.

3. Anomaly Patterns

Anomalies are detected based on a combination of feature values, including:

a. Negative to very high entropy values.
b. Unusual combinations of incoming and outgoing packet counts.
c. Extremely short or negative session durations.

3.6. Model Evaluation

The performance of the model was evaluated using several key metrics. The following table summarizes the evaluation results :

Model Performance Evaluation :

	Metrics	Value
0	Akurasi	0.429206
1	Presisi	0.143705
2	Recall	0.028749
3	F1-Score	0.047912

Figure 6. *Evaluation metrics*

The figure 6 shown evaluates the performance of the anomaly detection model using several key metrics. Here is an explanation of the results :

a. Accuracy : 0.429206

The model correctly classified roughly 43% of instances. However, accuracy alone is not sufficient for evaluating the performance of models dealing with imbalanced datasets.

Jurnal Teknik Informatika Vol. 18 No. 1, April 2025 (77-86)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v18i1.44285

b. Precision : 0.143705

Only a small proportion of the predicted anomalies were true anomalies, indicating a high number of false positives.

c. Recall : 0.028749

The model identified only 2.87% of the actual anomalies, showing its limited ability to detect rare events.

d. F1-Score : 0.047912

The model's F1-score is low, reflecting poor performance in balancing precision and recall.

Given the low performance across these metrics, further tuning of the model's parameters, exploration of additional feature engineering techniques, or even trying alternative algorithms is necessary to improve its anomaly detection capabilities.

3.7. Correlation Analysis

A correlation matrix was generated to analyze the relationships between the features. This matrix helps identify the most influential features in detecting anomalies. Strong correlations between bytes_in and bytes_out, for example, suggest that these features can provide critical insights into network traffic patterns.

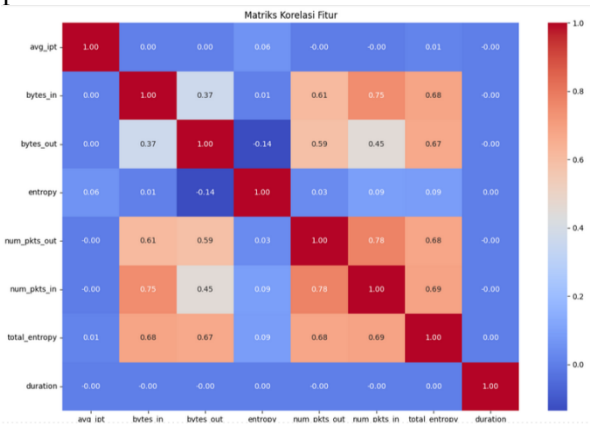


Figure 7. *Correlation matrix*

Figure 7 visualizes the correlation matrix, revealing strong relationships between several features. For example, the high correlation between bytes_in and bytes_out indicates a normal bidirectional flow of data. A sudden drop in this correlation may point to suspicious activities like DDoS attacks, which the Isolation Forest algorithm is designed to detect.

The correlation matrix illustrates the linear relationships between features in the dataset. The correlation coefficient values range from -1 to 1 :

a. Positive values indicate a positive correlation, meaning that as one feature increases, the other tends to increase as well.

b. Negative values indicate a negative correlation, meaning that as one feature increases, the other tends to decrease.

c. Values close to 0 indicate a weak correlation or no linear relationship between the two features.

Feature Analysis and Correlation Values in Figure 7 :

a. avg_ipt (Average Inter-Packet Time)
   1. Strong positive correlation with duration: Indicates that longer flow durations tend to have larger average inter-packet times. This makes sense as longer durations allow more time between individual packets.
   2. Moderate positive correlation with num_pkts_in and num_pkts_out: Suggests that flows with more packets tend to have larger inter-packet times.

b. bytes_in (Incoming Bytes)
   1. Very strong positive correlation with bytes_out, total_entropy, and num_pkts_in: Indicates that flows with more incoming bytes also tend to have more outgoing bytes, higher total entropy, and more incoming packets. This is consistent with normal network traffic behavior, where bidirectional data exchange involves significant data transfer.
   2. Moderate positive correlation with duration and num_pkts_out: Suggests that flows with more incoming bytes tend to have longer durations and more outgoing packets.

c. bytes_out (Outgoing Bytes)
   1. Very strong positive correlation with bytes_in, total_entropy, and num_pkts_out: Highlights a close relationship between outgoing bytes, total entropy, and the number of outgoing packets.
   2. Moderate positive correlation with duration and num_pkts_in: Indicates that flows with more outgoing bytes

tend to have longer durations and more incoming packets.

d. entropy (Entropy)
   1. Strong positive correlation with total_entropy: Suggests a strong relationship between flow entropy and total entropy.
   2. Moderate positive correlation with bytes_in, bytes_out, num_pkts_in, and num_pkts_out: Indicates that flows with higher entropy tend to have more incoming and outgoing bytes, as well as more incoming and outgoing packets.

e. num_pkts_in (Number of Incoming Packets)
   1. Very strong positive correlation with bytes_in, total_entropy, and num_pkts_out: Highlights a close relationship between the number of incoming packets, incoming bytes, total entropy, and the number of outgoing packets.
   2. Moderate positive correlation with duration and avg_ipt: Suggests that flows with more incoming packets tend to have longer durations and larger inter-packet times.

f. num_pkts_out (Number of Outgoing Packets)
   1. Very strong positive correlation with bytes_out, total_entropy, and num_pkts_in: Indicates a close relationship between the number of outgoing packets, outgoing bytes, total entropy, and the number of incoming packets.
   2. Moderate positive correlation with duration and avg_ipt: Suggests that flows with more outgoing packets tend to have longer durations and larger inter-packet times.

g. total_entropy (Total Entropy)
   1. Very strong positive correlation with bytes_in, bytes_out, num_pkts_in, and num_pkts_out: Suggests that total entropy is highly related to the number of bytes and packets both incoming and outgoing.
   2. Strong positive correlation with entropy: Indicates that total entropy is closely tied to flow entropy.

3.8. Future Directions

For future work, several avenues for improvement can be pursued :

a. Hyperparameter Tuning : Fine-tuning the Isolation Forest parameters using Grid Search or Random Search may improve performance.
b. Integration with Real-Time Systems: Implementing the model in real-time network monitoring systems could help in immediate anomaly detection, providing a proactive defense against cyberattacks.
c. Comparison with Other Algorithms: Although Isolation Forest performed well, comparing it with other anomaly detection algorithms, such as One-Class SVM or Autoencoders, could provide insights into alternative approaches.

CONCLUSION

This study investigates the application of the Isolation Forest algorithm for anomaly detection in network traffic, utilizing the Luflow Network Intrusion Detection dataset.

Anomaly Detection Capability, The Isolation Forest algorithm effectively detects anomalies in network traffic by isolating distinct data points. Its ability to handle high-dimensional and unlabeled datasets makes it particularly suitable for dynamic network environments, where features can vary and evolve.

Feature Selection and Preprocessing Impact, Key features such as bytes in, bytes out, entropy, and duration have a significant impact on model performance. Proper normalization and encoding techniques are essential to enhance the model's capacity to distinguish between normal and anomalous traffic patterns.

Model Performance and Areas for Improvement, While the algorithm shows potential, the model's performance metrics, including accuracy, precision, recall, and F1 score, highlight areas for improvement. Specifically, the recall of 2.87% and F1 score of 4.79% indicate challenges in detecting true anomalies and minimizing false positives. These issues can be addressed through hyperparameter tuning and alternative feature engineering methods.

Potential for Enhancing Network Security, The results demonstrate the potential of using machine learning algorithms like Isolation Forest to enhance network security. By identifying potential threats such as Distributed Denial of Service (DDoS) attacks, the model can be integrated into proactive anomaly detection systems that improve network defense mechanisms.

Future Directions, To optimize the model's efficiency, future work should focus on hyperparameter tuning, incorporating real-time detection capabilities, and comparing the Isolation Forest algorithm against other anomaly detection methods such as One-Class SVM or Autoencoders. Additionally, an in-depth error analysis, especially concerning false positives and negatives, would provide valuable insights into improving detection accuracy. Further evaluation of real-world challenges, such as handling imbalanced datasets and minimizing false positives, will also be crucial for practical deployment in network security systems.

REFERENCES

[1] B. Arifwidodo, Y. Syuhada, and S. Ikhwan, "Analisis Kinerja Mikrotik Terhadap Serangan Brute Force Dan DDoS," *Techno.Com*, vol. 20, no. 3, pp. 392–399, 2021, doi: 10.33633/tc.v20i3.4615.

[2] T. Tan and B. Soewito, "Manajemen Risiko Serangan SiberMenggunakan Framework NistCybersecurity Di Universitas Zxc," *J. Inf. Syst. Applied, Manag. Account. Res.*, vol. 6, no. 2, pp. 411–422, 2022, doi: 10.52362/jisamar.v6i2.781.

[3] S. Wang, J. F. Balarezo, S. Kandeepan, A. Al-Hourani, K. G. Chavez, and B. Rubinstein, "Machine learning in network anomaly detection: A survey," *IEEE Access*, vol. 9, pp. 152379–152396, 2021, doi: 10.1109/ACCESS.2021.3126834.

[4] S. Situmorang and Yahfizham, "Analisis Kinerja Algoritma Machine Learning Dalam Deteksi Anomali Jaringan," *J. Mat. dan Ilmu Pengetah. Alam*, vol. 1, no. 4, pp. 258–269, 2023, [Online]. Available: https://doi.org/10.59581/konstanta.v1i4.1722

[5] Gregorius Hendita, "Sistem Firewall untuk Pencegahan DDOS ATTACK di Masa Pandemi Covid-19," *J. Informatics Adv. Comput.* , vol. 3, no. 1, pp. 52–56, 2022, [Online]. Available: https://journal.univpancasila.ac.id/index.php/jiac/article/view/3853

[6] M. A. -, E. I. Alwi, and I. As'ad, "Analisis Forensik Terhadap Serangan Ddos Ping of Death Pada Server," *Cyber Secur. dan Forensik Digit.*, vol. 5, no. 1, pp. 23–31, 2022, doi: 10.14421/csecurity.2022.5.1.3423.

[7] Rakhmadi Rahman, Andi Maharani, and Nur Azisah Basir, "Detektor Anomali Jaringan dengan Analisis Perilaku untuk Mengidentifikasi Ancaman Persisten," *SABER J. Tek. Inform. Sains dan Ilmu Komun.*, vol. 2, no. 4, pp. 01–11, 2024, doi: 10.59841/saber.v1i3.1596.

[8] R. M. Imam, P. Sukarno, and M. A. Nugroho, "Deteksi Anomali Jaringan Menggunakan Hybrid Algorithm," *e-Proceeding Eng.*, vol. 6, no. 2, pp. 8766–8787, 2019, [Online]. Available: https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/viewFile/9868/9727

[9] J. Lesouple, C. Baudoin, M. Spigai, and J.-Y. Tourneret, "Generalized isolation forest for anomaly detection," *Pattern Recognit. Lett.*, vol. 149, pp. 109–119, Sep. 2021, doi: 10.1016/J.PATREC.2021.05.022.

[10] M. Carletti, M. Terzi, and G. A. Susto, "Interpretable Anomaly Detection with DIFFI: Depth-based feature importance of Isolation Forest," *Eng. Appl. Artif. Intell.*, vol. 119, p. 105730, Mar. 2023, doi: 10.1016/J.ENGAPPAI.2022.105730.