JIKO (Jurnal Informatika dan Komputer)
Vol. 6, No. 2, Agustus 2023, hlm. 79-87
DOI: 10.33387/jiko.v6i2.6092

Accredited KEMENDIKBUD RISTEK, No.105/E/KPT/2022
p-ISSN: 2614-8897
e-ISSN: 2656-1948

DATA MINING IMPLEMENTATION FOR DETECTION OF ANOMALIES IN NETWORK TRAFFIC PACKETS USING OUTLIER DETECTION APPROACH

Kurnia Setiawan[1], Arief Wibowo[2]

[1]Politeknik Siber dan Sandi Negara
[1,2]Universitas Budi Luhur
[*]Email: [1]1911601324@student.budiluhur.ac.id, [2]arief.wibowo@budiluhur.ac.id

(Received: 11 May 2023, Revised: 224 May 2023, Accepted: 5 June 2023)

Abstract

A large number of data packet records of network traffic can be used to evaluate the quality of a network and to analyze the occurrence of anomalies in the network, both related to network security and performance. Based on the data obtained, the event of irregularities in computer networks can not be explicitly detected on which traffic packets. Meanwhile, monitoring network traffic packets manually will require a lot of time and resources, making it difficult to see potential anomaly events more precisely. This study analyzes network packet traffic data through several general steps, including the process of capturing network packet traffic data, identifying security vulnerabilities by searching for anomalous traffic packets using an outlier detection approach, validating the results of outlier detection with expert information, and then conducting a classification and evaluation process. In the outlier detection process using the Isolation Forest algorithm, the results showed 1,643 records (4.86%) classified as outlier types, while 32,098 (95.13%) classified as inliers. Subsequently, validating and filtering the expert attributes containing expert information is conducted. The outlier detection results were classified using five comparison algorithms: Random Forest Classifier, Support Vector Machine, Decision Tree Classifier, K-Nearest Neighbor, and Bernoulli Naive Bayes. The Random Forest algorithm has the highest score for accuracy, macro average precision, and macro average f1-score, namely 0.9962067330488383, 0.78, and 0.82.

Keywords: *Traffic packets, Anomalies, Isolation Forest algorithm, Random Forest algorithm*

**Corresponding Author: Kurnia Setiawan*

1. INTRODUCTION

Computer networks are essential for preserving an organization business processes's continuity, particularly with the fast-paced development of the internet and interconnecting technologies. To ensure the availability of computer network services, one must take steps to eliminate threats in network traffic. Network traffic refers to the data present on the network when it is active, with information being exchanged in the form of service resource access requests and provider responses [1]. Threats to network traffic can come from several sources, such as malicious activities utilizing network services, network overload, device damage, and the leak of network parameters such as protocols and ports [1]. To address these threats, it is necessary to monitor network traffic using a monitoring system to capture, analyze, and determine whether a packet is normal or an anomaly. Traffic anomalies in the network are defined as events or operations that deviate from the normal behavior of the network [1].

Based on network monitoring data from October 2021, 1,075 anomalous occurrences were found in computer networks involving 187 devices, but the specific traffic packets involved have not been detected. To carry out further analysis and investigation, it is necessary to gather more specific information about the anomalous traffic packets. This is crucial for investigating security and network performance issues related to these anomalies. However, manual monitoring of traffic packets would require a significant amount of time and resources, making it difficult to detect potential anomalous events in network traffic packets with specificity.

There are two commonly used approaches in monitoring systems for detecting threats in a network: Misuse detection and Anomaly-based detection [2]. Misuse detection systems, also known as Signature-based detection, use a recognized pattern or a database

of suspect activities and operations that may be dangerous. However, this approach has limitations, as it can only detect intrusions whose attack patterns have already been identified and stored. New and unrecognized attack patterns are less detectable. Anomaly-based detection aims to recognize new attack patterns, assuming that every intrusion activity is an anomaly in the network, and builds a normal activity profile [1]. Outlier Detection is a technique used to detect network traffic anomalies, where the assumption is that an anomaly is an unusual occurrence in the network [1]. This research focuses on using data mining methods to detect potential anomalies in network traffic packets and classify them more accurately.

In the related previous research [3], it was found that the Isolation Forest algorithm had the highest accuracy and F1 score, specifically at a flow-timeout threshold of 120s, with values of 98% and 87% respectively. However, the True Positives (TP) rate was not very high, reaching only 77%. This means that the correct classification of successfully detected malicious flows is not very high. In addition, the research conducted by [4] utilized Autoencoder (AE) and Isolation Forest (IF) for the binary classification of incoming traffic on fog devices, aiming to make real-time decisions on attacks and normal traffic. The study achieved a high accuracy rate of 95.4%. However, it should be noted that the classification process is conducted in real time without validation through expert information, which could potentially enhance the accuracy of detection. Therefore, in this study, further development was conducted using a different dataset and by incorporating expert information validation methods during outlier detection. This ensures that outlier detection results are not immediately labeled as anomalies. Subsequently, the detected anomalies were classified using the Random Forest algorithm, resulting in an accuracy of 99%. Each label also had its own accuracy, precision, recall, and F1 score values.

The data used in this study was obtained from network traffic packets captured using Wireshark. The data mining method with the Outlier Detection algorithm approach uses the Isolation Forest algorithm. Subsequently, the detected anomalies are classified using a data mining classification algorithm, enabling the development of a classification model for detecting packet traffic anomalies on the network. This allows the models and prototypes developed to be utilized for in-depth analysis of network traffic packet data, thereby detecting potential anomalous occurrences on the network.

2. RESEARCH METHOD

This research was conducted to determine the most appropriate data mining model for detecting potential network anomalies based on network traffic packet datasets. This is because there are difficulties in conducting an in-depth analysis of packet capture network traffic, and it is challenging to manually detect anomalies with a large number of traffic packets. In related studies, the recommended data mining classification model for detecting network anomalies in network traffic packet datasets is not explicitly explained. In the case of commonly used data mining classification methods for detecting anomalies in public NSL-KDD datasets, there are several research references related to the classification of Intrusion Detection System (IDS) data. One such study, conducted by [5], compares the Naive Bayes (NB), Random Tree, Random Forest (RanFor), and J48 methods for classification. The results showed that, in comparison to other algorithms, the RF approach offered a relatively higher level of accuracy [5]. This study is relevant to the review as research [6] and [4] have similarities, using the Isolation Forest algorithm to detect outliers or anomalies in fog computing devices and network management systems. There is also relevance to other research related to network traffic capture data, such as research [7] that compares the Decision Tree (DT), Support Vector Machine (SVM), K-Nearest Neighbor (K-NN), and Convolutional Neural Network (CNN) methods on network packet capture data (using the KDD NSL dataset) to detect Distributed Denial of Service attacks.

Previous research related to outlier detection can be seen in [8], which compares outlier detection in two datasets of hospital drug use through a comparison between the K-Means, LOF, and OC-SVM algorithms. Another example can be seen in [9], where the researchers utilized the K-Means clustering algorithm and neural networks to detect outliers in social media network analysis. The Isolation Forest algorithm was chosen for outlier detection in this study due to its ability to eliminate computational costs by not relying on distance or density measures. This advantage makes the Isolation Forest algorithm a suitable choice for detecting anomalies in large and complex data sets [4]. The Isolation Forest algorithm adopts a unique approach to anomaly detection by isolating instances without relying on distance or density, capitalizing on two key properties of anomalies: they are a minority of instances and they have attribute values that significantly differ from those of normal examples, or, in other words, anomalies are "few and different" [10].

Previous studies have used the Isolation Forest algorithm in various ways. In [4], the Autoencoder (AE) and the Isolation Forest (IF) algorithms were used for binary classification of incoming traffic on fog devices, where the system had to make real-time decisions regarding attacks and normal traffic. Another study [6] employed statistical methods to extract features and utilized the Isolation Forest algorithm to perform two classifications, resulting in fast and accurate anomaly detection across various types of data. Other research related to data mining classification conducted by [5], [11], [12], [13], [14], [15], [16], [17], [18], [19], and [20] have used

algorithms such as SVM, K-NN, General Regression Neural Network, RanFor, C4.5, NB, and analysis of visitor logs with average errors. The research utilizes stages of method the Cross-Industry Standard Process for Data Mining (CRISP-DM), as outlined in [21], as a guide for incorporating data mining into problem-solving strategies in businesses or research units. In general, the steps taken in this research include data collection and data preparation, outlier detection and validation, classification and evaluation, as well as prototype design, as depicted in Figure 1.

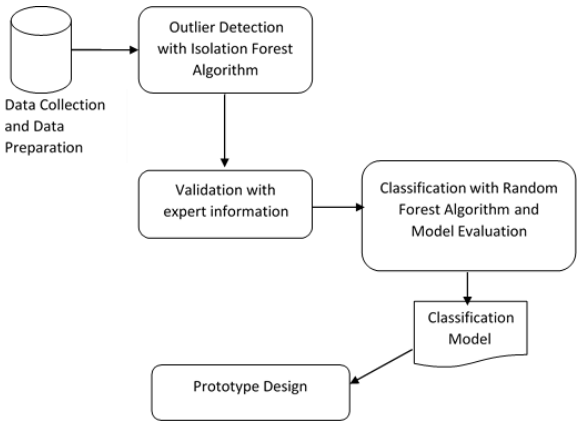

Figure 1 Data Analysis Diagram of Potential Anomaly Detection Modeling

The explanation for Figure 1 is as follows, depicting the sequential stages of the method: Data Collection and Data Preparation, Outlier Detection and Validation, Classification and Evaluation, and Prototype Design.

2.1 Data Collection and Data Preparation

The data collection method in this study involves capturing network traffic packets using Wireshark tools, which are open-source tools for packet capturing. Wireshark provides more in-depth information and expert insights to assist in the process of detecting outliers and classifying potential network traffic packet anomalies.

Based on the packet capture data from the network, the data is prepared for processing in the next step. The data preparation process involves setting the columns that appear in the Wireshark application, saving the network traffic packet data file in CSV format, selecting attributes (columns) for the outlier detection process, and undergoing several other data preparation steps.

2.2 Outlier Detection and Validation

Outlier analysis is performed to detect potential anomalies in network traffic packets using the Isolation Forest method. This method separates entities that are outliers (few and different) from normal entities by randomly selecting two features from the network traffic packet dataset and comparing the data with the range of values of the selected features to isolate them.

After performing the outlier detection, the results were validated by examining the expert information. The expert information includes "Note", "Chat", "Warning", and "Error" information, which provides initial guidance for the investigation. Expert information serves as a starting point to give a clearer picture of unusual or significant network behavior. It's essential to remember though, that the existence of expert information does not inevitably signify a problem and the absence of expert information does not always mean everything is functioning normally [22]. To get a more specific number of outliers, the results of the outlier detection were further analyzed by filtering the outliers based on the expert information in the form of "Warning" and "Error". These two statements indicate serious warnings and problems, making the data more likely to be anomalous and a priority for further investigation.

2.3 Classification and Evalution

After obtaining the new labels, including normal data labels/inliers, outliers, errors, and warnings & outliers, the next step is to perform data mining classification to obtain a classification model for potentially anomalous network traffic packets. The classification process was performed using five different algorithms: Random Forest, Support Vector Machine, Decision Tree, K-Nearest Neighbor, and Bernoulli Naive Bayes. An evaluation was then conducted by comparing the scores of accuracy, precision, recall, and F1-score.

The Random Forest algorithm has the highest accuracy score compared to other algorithms, so the implementation of the Random Forest algorithm is broken down into the following algorithm stages:

a. Bootstrap dataset creation
   An estimation technique known as bootstrap is used to produce the predictions, by resampling the data set and selecting a random sample from the initial data set using a bagging process.
b. Making a Decision Tree (Decision tree)
   By using the bootstrap dataset produced in the preceding step, construct a decision tree. Additionally, when creating it, just a portion of the dataset is taken into account, with a subspace of random variables being chosen as root nodes at each stage. Then repeat the same process for each subsequent branch node, by randomly selecting several variables as candidate branch nodes.
c. Return to the initial step and repeat
   Each decision tree in Random Forest categorizes output classes according to the specific predictor factors that were employed. The output class is determined by which class receives the most votes.

After obtaining the classification results with the highest-scoring algorithm, the classification model is stored in the Pickle format. A simple application is then developed using the Django framework for

classifying potentially anomalous network traffic packet data.

2.4 Prototype Design

The development of the application uses the Pickle model that was generated and stored in the modeling stage, using the Python programming language with the Django frame in the development of application prototype. Thus, if there is new network traffic packet data or additional data, modeling can be carried out again to produce an updated model, which is expected to result in more accurate classification.

3. RESULT AND DISCUSSION

Based on the network traffic packet capture data obtained, there are 33,741 traffic packet records captured with attributes consisting of number, time (time in seconds), source (source IP address), destination (destination IP address), protocol (network protocol used). used), length (packet length in bytes), and info (explanation of information). To get more complete data, a simple configuration is performed on the traffic packet capture results, in columns preferences to display other information, namely source port, destination port, stream index, TCP segment. len (tcp segment length), sequence number (tcp sequence), sequence number (raw), next sequence number, acknowledgment number, acknowledgment number (raw), header length (tcp header length), flags (tcp flags), and expert (expert information in the form of hints according to severity level). After that, the attributes that will be used for the data mining process are selected, namely: protocol, length, source port, destination port, TCP segment len, sequence number, acknowledgment number, header length, flags, and expert. These data attributes have the characteristics of different data types, categorical data types are found in the attributes: protocol, flags, and expert, while numeric data types are found in the attributes: length, source port, destination port, TCP segment len, sequence number, acknowledgment number, and header length.

It was found that the most used Protocol Attribute was TCP Protocol with 18,594 instances. TCP Protocol is a protocol used for exchanging information between devices on the network. The initial data contained records with null or NaN (Not a Number) values, which required several data preparation steps before the first data mining process using the Isolation Forest algorithm could be performed. The first step involved transforming the null or NaN data. In addition, categorical data, such as protocol, flags, and expert, were transformed into numeric data.

3.1 Outlier Detection and Validation

The first modeling implementation to detect outliers from network traffic packet data uses the Isolation Forest algorithm. The steps taken are as follows:

a. Determine the x variable which consists of the protocol, length, source port, destination port, tcp segment length, sequence number, acknowledgment number, header length, and flags attributes, and the y variable is the expert attribute.
b. Calculates the isolation forest score for each record.
c. Mark a score with a minus (-) value with an anomaly label -1 and vice versa with a value of 1. Then each label -1 is added with outliers and label 1 is added with inliers. Graphical display of the number of outliers and inliers data frequencies can be seen in the following figure.

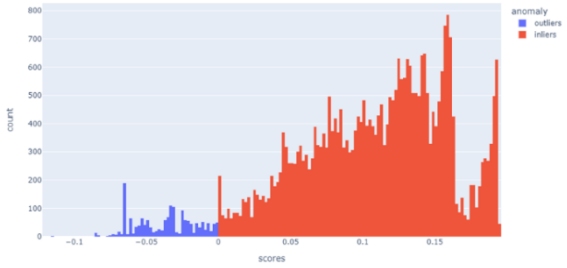

Figure 2 Histogram Data Outliers and Inliers

In Figure 2 it is explained that based on these results, 1,643 records (4.86%) were found to be outliers, while inliers were 32,098 records (95.13%). A check is made on the attribute of the expert which contains expert information which is the starting point of the investigation, to give a better picture of unusual or important network behavior, but the presence of expert information does not always indicate a problem and there is no expert information doesn't always mean all is well. The expert information contained in the traffic packet data capture consists of 4,231 records labeled "Note", 1,538 "Warning", 1,051 "Chat", and 15 "Error". The description "Chat" is information about ordinary workflows, "Note" is a note of important events, "Warn" is a warning when an application gives an odd error code, such as one indicating a connection issue, and "Error" is a serious problem, such as a defective package (malformed). So that the outlier detection data is filtered which has expert information in the form of "Warn" and "Error" and with the description "outlier", to label the potential anomaly data.

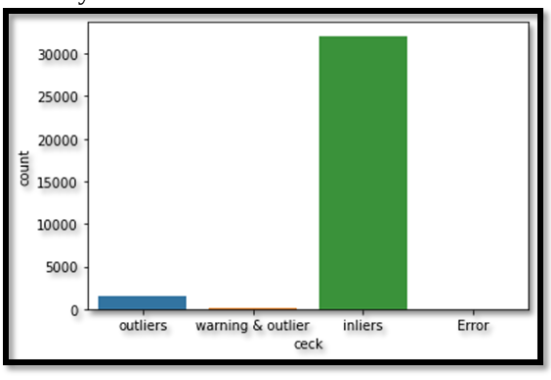

Figure 3 The amount of data labeled "inliers", "outliers", "warning & outlier", and "Error"

In Figure 3 it is explained that the number of potential anomaly data labels consists of 89 records with the description "warning & outlier", 15 "error", 1,554 "outliers", and 32,083 "inliers".

3.2 Classification and Evalution

Based on the results of the outliers detection, classification was carried out using 5 algorithms as a comparison, where the RanFor algorithm had the highest accuracy score. In Table 1 it is explained that comparison of the accuracy scores of each algorithm in the classification process can be seen as follows.

Table 1 comparison of classification algorithm accuracy scores

No.	Algorithms	Accuracy Score
1.	Random Forest	0,9962067330488383
2.	Support Vector Machine	0,9467757230915126
3.	Decision Tree	0,9958511142721669
4.	K-Nearest Neighbor	0,9770033191085823
5.	Bernoulli Naive Bayes	0,9468942626837363

3.2.1 Classification process

The classification process in the Random Forest algorithm can be explained as follows,

a. Bootstrap dataset creation

Bootstrapping is a technique that involves picking random samples from the original data set using a bagging process in order to generate predictions on data sets.

- Defined index to divide the dataset into training data and testing data

BEGIN
NUMERIC population, k, test_indices, ts_d, tn_d
population = data index
k = test_size
test_indices = random sampling of the population with size k
ts_d = data selection in test_indices from row to row and from column to column
tn_d = remaining data other than ts_d
END

- Also defined is the bootstrapping process to get a random sample and replace it

BEGIN
NUMERIC tn_d, bootstrap_indices, n_btp, df_bootstrap
bootstrap_indices = random sampling on tn_d by returning an array and filling it with random integers in low (0) to high (length of tn_d) intervals
df_bootstraped = selection of data that is bootstrap_indices in tn_d
END

- Bootstrapping process

b. Conduct training dataset on a number of random features

- Define a function to get the split potential of the training data

BEGIN
NUMERIC data, random_subspace, n_columns, columns_indices, k, unique_values, potential_splits
n_columns = number of columns in the data
column_indices = value data in the range of all features except labels/classes (n_columns – 1)
IF random_subspace & random_subspace <= number of column_indices THEN
column_indices = random sampling on the column_indices population with size k = random_subspace
FOR column_indices
unique_values = defines the unique elements of the data
the potential_splits output of column_index is unique_values
END FOR
END

- Do split data training with random subspace
- The Random Forest algorithm is defined, with the parameter tn_d being the dataset, n_tri being the number of trees, n_btp being the size of the bootstrap dataset, n_fs being the number of features to be trained, and dt_md is the maximum tree depth

BEGIN
NUMERIC tn_d, n_tri, n_btp, n_fs, dt_md
n_tri = number of trees
n_btp = bootstrap dataset size
n_fs = number of features to be trained
dt_md = maximum tree depth
FOR i on range n_tri
d_btpd = selection of data which is bootstrap_indices in tn_d with bootstrap dataset size n
tree = forming a tree using the d_tri algorithm, with d_btpd data selection, dt_md tree depth, and random_subspace a number of features trained
output forest = adds the tree that is formed in the list that contains all trees
END FOR
END

- Training is carried out using the Random Forest algorithm

BEGIN
NUMERIC tn_d, n_tri, n_btp, n_fs, dt_md, random_forest_algorithm
n_tri = number of trees
n_btp = bootstrap dataset size
n_fs = number of features to be trained
dt_md = maximum tree depth
input n_tri, n_btp, n_fs, dt_md
forest = random_forest_algorithm training process with dataframes tn_d, n_tri, n_btp, n_fs, and dt_md
END

c. Perform classification with the Random Forest algorithm

BEGIN
NUMERIC ts_d, forest, df_predictions
d_tri_predictions = sample prediction process
FOR i in the long range of the forest
df_predictions = process of predicting d_tri_predictions with ts_d and a number of tree forests i
END FOR
END

Classification process with iterations 10 times by also evaluating the results of the average accuracy score.

BEGIN
NUMERIC tn_d, ts_d, n_tri, n_btp, n_fs, dt_md, forest, predictions, accuracy
input n_tri, n_btp, n_fs, dt_md
FOR i in range 10
forest = random_forest_algorithm training process with dataframes tn_d, n_tri, n_btp, n_fs, and dt_md
predictions = process predictions random_forest_predictions with ts_d and forest a number of i
accuracy = calculates the accuracy of the prediction results for the ts_d label
END FOR
accuracy score output for each i and average accuracy
END

d. Evaluation of classification results

In table 2 it is explained that based on the classification process, the confusion matrix is obtained as follows.

Table 2 Precision, recall, f1-score, and support for the Random Forest algorithm

	Precision	Recall	F1-score	Support
Error	0,00	0,00	0,00	0
Inliers	1,00	0,99	0,99	6503
Outliers	0,76	0,90	0,82	245
Warning & Outlier	0,00	0,00	0,00	0
Accuracy			0,99	6748
Macro avg	0,44	0,47	0,45	6748
Weighted avg	0,99	0,99	0,99	6748

Based on the table it is known that there are 2 classes/labels that are not classified, this may be due to imbalanced data, where the labels "Error" and "Warning & Outlier" have a relatively small number of records when compared to other labels.

e. Perform over sampling using the SMOTE method

Oversampling is done by resampling using the SMOTE method, namely synthesizing new samples from the minority class to balance the dataset by resampling the minority class samples.

After oversampling, the number of labels produced is illustrated in Figure 4 below.

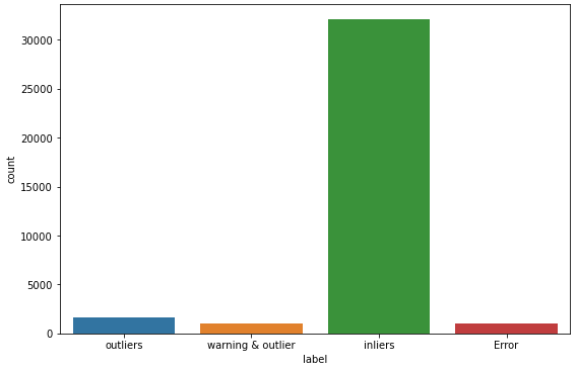

Figure 4 Comparison of the number of records on each label after over sampling

Based on the oversampling results, a classification process is carried out with the The Random Forest algorithm and the confusion matrix are obtained as described in table 3 below.

Table 3 Precision, recall, f1-score, and support with Random Forest based on oversampling

	Precision	Recall	F1-score	Support
Error	0,38	0,94	0,54	78
Inliers	0,99	0,96	0,97	6685
Outliers	0,57	0,74	0,65	230
Warning & Outlier	0,69	0,92	0,79	154
Accuracy			0,95	7147
Macro avg	0,66	0,89	0,74	7147
Weighted avg	0,96	0,95	0,95	7147

3.2.2 Characteristics of each classification label

Based on these results, it is known that all classes/labels can be classified, but there are labels with not too high precision, recall, and f1-score, namely:

a. The “Error” label has a precision score of 0.38; this shows that the ratio of true positive classification compared to the overall results classified as positive is less than half, so that the percentage of correct labels "error" of all those classified as "error" is quite low. Recall score 0.94; shows the ratio of true positive classification compared to all data that is true positive is high, so that the percentage classified as "Error" compared to all data that is actually labeled "Error" is high. And the f1-score of 0.54 shows that precision and recall's overall mean is not particularly high.

b. The "Outliers" label has a precision score of 0.57; this shows that the ratio of true positive classification compared to the overall results classified as positive is half, so that the percentage of correct labels "Outliers" of the total classified as "error" reaches half. In addition recall score of 0.74, indicating the ratio of true positive classification compared to all data that is true positive is sufficient but not too high, so that the percentage classified as "Outliers" compared to all data that is actually labeled "Outliers" is sufficient

but not too high. And the f1-score 0.65 indicates that precision and recall's overall mean is not particularly high.

c. The “Warning & Outlier” label has a precision score of 0.69; this shows that the ratio of true positive classification compared to the overall results that are classified as positive is sufficient but not too high, so that the percentage of correct labels "error" from all those classified as "Warning & Outlier" is sufficient but not too high. Recall score 0.92; shows the ratio of true positive classification compared to all data that is true positive is high, so that the percentage classified as "Warning & Outlier" compared to all data that is actually labeled "Warning & Outlier" is high. And the f1-score of 0.79 shows that precision and recall's overall mean is not particularly high.

3.3 Prototype Design

Based on the classification results achieved by the Random Forest algorithm, deployment is carried out for building a simple application prototype to classify whether or not there is a potential anomaly from network traffic packet data, using the Random Forest classification model that has been stored. This simple application is to confirm that the model that has been formed can be implemented for classifying traffic packet data. The development of the application uses the Django framework.

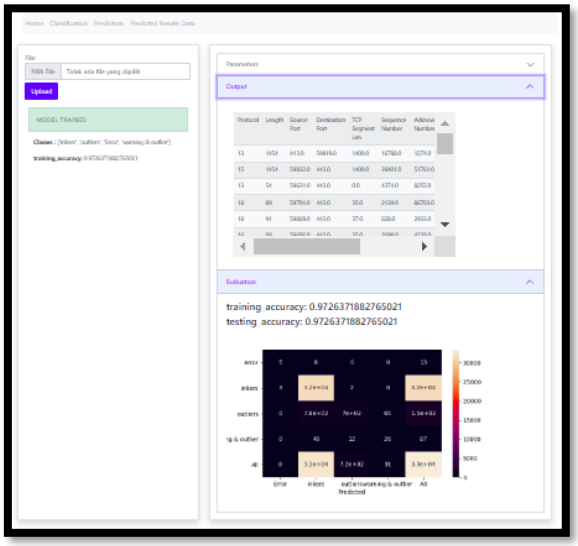

Figure 5 Classification process with the RanFor algorithm on the prototype

The classification page describes the functions for uploading and selecting the dataset to be used. Furthermore, there is a classification function using the Random Forest algorithm, which produces an accuracy score and a confusion matrix as an evaluation, as illustrated in the screenshot of the application in figure 5. On the prediction classification page, new traffic packet data is entered as input for classification as a potential anomaly or not, using the Random Forest algorithm classification results model, illustrated in the screenshot of the application in figure 6.

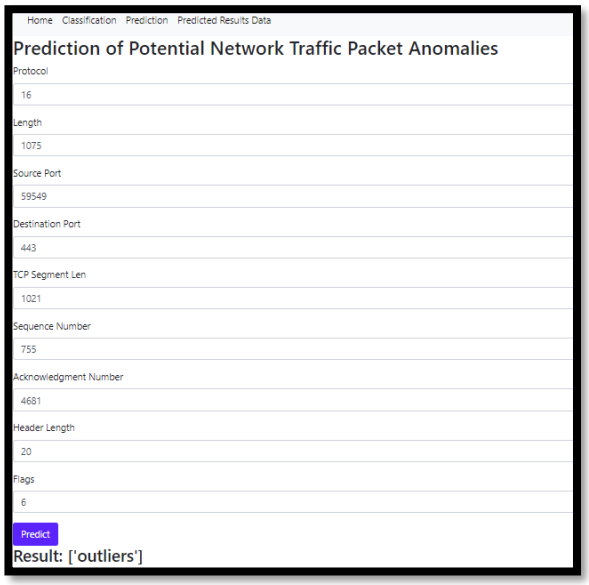

Figure 6 Classification results of traffic packet data with the label "Outliers"

Besides that, on the Classification Results Data page, users can see the history of the process of classifying data traffic packets that can be exported, illustrated in the screenshot of the application in figure 7.

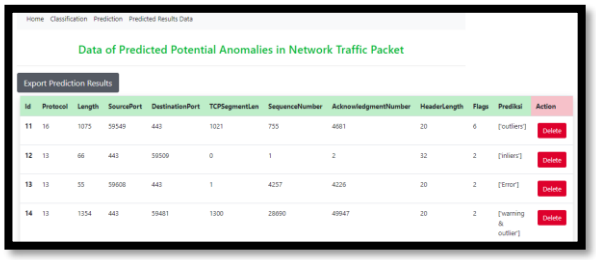

Figure 7 Classification result data

Based on the results of these studies, a data mining algorithm with an outlier detection approach, Isolation Forest algorithm, can be implemented to detect potential network traffic packet anomalies, and the RanFor algorithm can be implemented to accurately classify potential network traffic packet anomalies.

4. CONCLUSION

In this research, an anomaly detection and potential anomaly classification were carried out on network traffic packet data using the Isolation Forest algorithm for outlier detection and the Random Forest algorithm for classification. The Isolation Forest algorithm identified 1,643 records (4.86%) as outliers and 32,098 records (95.13%) as inliers. The results of the outlier detection were then checked against the "expert" attribute, which contains expert information and serves as a starting point for investigations. Outliers with expert information labeled as "Warn" and "Error" were filtered and labeled as potential

anomaly data. The potential anomaly data was then labeled as 89 records with "warning & outlier", 15 with "error", 1,554 as "outliers", and 32,083 as "inliers".

The Random Forest algorithm can be applied to classify potential network traffic anomalies. It was compared to five other classification algorithms, namely Random Forest Classifier, Support Vector Machine, Decision Tree Classifier, K-Nearest Neighbor, and Bernoulli Naive Bayes. Out of these five algorithms, the Random Forest algorithm showed the highest accuracy, macro average Precision, and macro average f1-score, with accuracy value 0.9962067330488383, precision macro average 0.78, and macro-average f1-score of 0.82. The macro average Recall was obtained from the Decision Tree Classifier, with a value of 0.93. Based on these findings, the Random Forest algorithm was chosen for classifying potential network traffic packet anomalies. The resulting classification model can categorize samples as "inliers", "outliers", "Error", and "warning & outlier". However, the "error" and "warning & outlier" labels have precision, recall, and f1-score values that are only sufficient but not high, with precision value 0.50, recall 1.00, and f1-score 0.67 for "error"; and precision value 0.64, recall 0.70, and f1-score 0.67 for "warning & outlier".

In this study, the potential anomalies in network traffic packet data have not been determined to be related to system performance or security. Further research, such as network security testing, is needed to confirm this. The findings of this study mainly categorize network traffic packet data that has the potential to be anomalous, labeled as "Error" and "Warning & Outlier." The classification model obtained from this study can aid in prioritizing the investigation process for anomalies, both related to performance and security. By starting with "Error" and "Warning & Outlier" data, the investigation can be further extended to "Outlier" and "Inlier" data if necessary.

For further research, the following avenues can be explored to improve the detection of network traffic anomalies:

- Using other methods such as those based on soft computing, knowledge-based, or a combination of learning algorithms.
- Combining intrusion detection with a misuse detection approach, which detects known attacks based on stored signatures, to make the anomaly detection process more comprehensive.
- Incorporating network security trials to confirm that the results of anomaly detection are related to system performance or security.

In addition, the development of a real-time integrated anomaly detection and classification system is also possible. This system would continuously improve its detection capabilities by continuously updating its models.

5. REFERENCE

[1] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, *Network Traffic Anomaly Detection Techniques and Systems*. Springer, 2017.

[2] V. Jyothsna and K. M. Prasad, "Anomaly-Based Intrusion Detection System," *Intechopen*, pp. 1–15, 2019.

[3] P. R. Grammatikis, P. Sarigiannidis, A. Sarigiannidis, D. Margounakis, A. Tsiakalos, and G. Efstathopoulos, "An Anomaly Detection Mechanism for IEC 60870-5-104," *2020 9th Int. Conf. Mod. Circuits Syst. Technol. MOCAST 2020*, pp. 0–3, 2020, doi: 10.1109/MOCAST49295.2020.9200285.

[4] K. Sadaf and J. Sultana, "Intrusion detection based on autoencoder and isolation forest in fog computing," *IEEE Access*, vol. 8, pp. 167059–167068, 2020, doi: 10.1109/ACCESS.2020.3022855.

[5] L. Mohan, S. Jain, P. Suyal, and A. Kumar, "Data mining Classification Techniques for Intrusion Detection System," *Proc. - 2020 12th Int. Conf. Comput. Intell. Commun. Networks, CICN 2020*, pp. 351–355, 2020, doi: 10.1109/CICN49253.2020.9242642.

[6] X. Chun-Hui, S. Chen, B. Cong-Xiao, and L. Xing, "Anomaly Detection in Network Management System Based on Isolation Forest," *Proc. - 2018 4th Annu. Int. Conf. Netw. Inf. Syst. Comput. ICNISC 2018*, pp. 56–60, 2018, doi: 10.1109/ICNISC.2018.00019.

[7] A. R. Shaaban, E. Abd-Elwanis, and M. Hussein, "DDoS attack detection and classification via Convolutional Neural Network (CNN)," *Proc. - 2019 IEEE 9th Int. Conf. Intell. Comput. Inf. Syst. ICICIS 2019*, pp. 233–238, 2019, doi: 10.1109/ICICIS46948.2019.9014826.

[8] E. H. Budiarto, A. Erna Permanasari, and S. Fauziati, "Unsupervised anomaly detection using K-Means, local outlier factor and one class SVM," *Proc. - 2019 5th Int. Conf. Sci. Technol. ICST 2019*, 2019, doi: 10.1109/ICST47872.2019.9166366.

[9] P. Kaur, "Outlier Detection Using Kmeans and Fuzzy Min Max Neural Network in Network Data," *Proc. - 2016 8th Int. Conf. Comput. Intell. Commun. Networks, CICN 2016*, pp. 693–696, 2017, doi: 10.1109/CICN.2016.142.

[10] F. T. Liu, K. M. Ting, and Z. H. Zhou, "Isolation-based anomaly detection," *ACM Trans. Knowl. Discov. Data*, vol. 6, no. 1, pp. 1–44, 2012, doi: 10.1145/2133360.2133363.

[11] D. Narayan, A. Malony, and C. Louella, "Intrusion Detection System Using Data Mining Techniques," *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 7, no. 5, pp. 450–452, 2017, doi: 10.23956/ijarcsse/v7i5/0161.

[12] K. Gurulakshmi and A. Nesarani, "Analysis of IoT Bots against DDOS attack using Machine

learning algorithm," *2018 2nd Int. Conf. Trends Electron. Informatics*, no. Icoei, pp. 1052–1057, 2018.

[13] H. Hafid, "Investigasi Log Jaringan Untuk Deteksi Serangan Distributed Denial of Service (Ddos) Dengan Menggunakan Metode General Regression Neural Network." 2019.

[14] M. O. Miah, S. S. Khan, S. Shatabda, and D. M. Farid, "Improving Detection Accuracy for Imbalanced Network Intrusion Classification using Cluster-based Under-sampling with Random Forests," *1st Int. Conf. Adv. Sci. Eng. Robot. Technol. 2019, ICASERT 2019*, vol. 2019, no. Icasert, pp. 1–5, 2019, doi: 10.1109/ICASERT.2019.8934495.

[15] Rastri Prathivi and Vensy Vydia, "Analisa Pendeteksian Worm dan Trojan pada Jaringan Internet Universitas Semarang menggunakan Metode Klasifikasi pada Data Mining," *J. Transform.*, vol. 14, no. 2, pp. 77–81, 2017.

[16] M. F. Fibrianda and A. Bhawiyuga, "Analisis Perbandingan Akurasi Deteksi Serangan Pada Jaringan Komputer Dengan Metode Naïve Bayes Dan Support Vector Machine (SVM)," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. II, no. 9, pp. 3112–3123, 2018.

[17] S. Anwar, F. Septian, and R. D. Septiana, "Klasifikasi Anomali Intrusion Detection System (IDS) Menggunakan Algoritma Naïve Bayes Classifier dan Correlation-Based Feature Selection," *J. Teknol. Sist. Inf. dan Apl.*, vol. 2, no. 4, p. 135, 2019, doi: 10.32493/jtsi.v2i4.3453.

[18] M. Nivaashini and P. Thangaraj, "A framework of novel feature set extraction based intrusion detection system for internet of things using hybrid machine learning algorithms," *2018 Int. Conf. Comput. Power Commun. Technol. GUCON 2018*, pp. 44–49, 2019, doi: 10.1109/GUCON.2018.8674952.

[19] 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.

[20] M. Reza Redo Islami, "Deteksi Dini Serangan Pada Website Menggunakan Metode Anomali Based Early Detection of Attacks on Websitesusing the Based Anomaly Method," *J. Inform. dan Komputer) Akreditasi KEMENRISTEKDIKTI*, vol. 5, no. 3, pp. 224–229, 2022, doi: 10.33387/jiko.

[21] D. T. and Larose and C. D. Larose, *Data Mining and Predictive Analytics*. JohnWiley &Sons,Inc., 2015.

[22] Wireshark, "Expert Information, Chapter 7. Advanced Topics." https://www.wireshark.org/docs/wsug_html_chunked/ChAdvExpert.html (accessed Jun. 24, 2022).