INNOVATIVE: Journal Of Social Science Research
Volume 4 Nomor 1 Tahun 2024 Page 10572-10588
E-ISSN 2807-4238 and P-ISSN 2807-4246
Website: https://j-innovative.org/index.php/Innovative

XGBoost Algorithm on Intrusion Detection System in Detecting Network Intrusions

Mutiara Hernowo[1✉], Endang Sugiharti[2]
Departement of Computer Science, Faculty of Mathematics and Natural Sciences, Universitas Negeri Semarang, Semarang, Indonesia
Email: mutiaraaj15@students.unnes.ac.id[1✉]

Abstrak

Saat ini, teknologi sudah menjadi kebutuhan manusia. Akibat peningkatan penggunaan internet, banyak paket data yang diteruskan ke lalu lintas jaringan tempat data berkomunikasi antara dua titik akhir (transmisi data). Aktivitas ini harus aman karena informasi pribadi pengguna bersifat rahasia. Jaringan memiliki sistem untuk menganalisis setiap data yang melewati lalu lintas dan mendeteksi data berbahaya, yang disebut Intrusion Detection System (IDS). IDS membutuhkan model deteksi untuk meningkatkan kinerjanya dalam mendeteksi intrusi. Tujuannya adalah untuk mengimplementasikan algoritma XGBoost untuk meningkatkan skor akurasi kinerja IDS menggunakan metode yang diusulkan. Dalam tulisan ini, kami mengusulkan model deteksi menggunakan algoritma XGBoost dan Sequential Feature Selection (SFS) sebagai metode pemilihan fitur. Metode-metode ini telah diuji pada dataset NSL-KDD. Melalui penelitian implementasi model yang diusulkan ini, diperoleh hasil dengan menganalisis metrik evaluasi seperti, akurasi, presisi, recall, dan f1-score. Hasilnya menunjukkan skor akurasi mencapai 99,24%. Dengan kata lain, hasilnya cukup tinggi dibandingkan penelitian sebelumnya. Dengan demikian, metode yang diusulkan dapat digunakan untuk meningkatkan kinerja IDS guna mendeteksi intrusi dan membantu jaringan menjadi lebih aman. Penelitian ini masih memerlukan pengembangan untuk penelitian selanjutnya karena teknologi terus berkembang.

Kata Kunci: *Deteksi intrusi, Intrusion Detection System (IDS), dataset NSL-KDD Sequential Feature Selection (SFS) Algoritma XGBoost*

Abstract

Nowadays, technology has become human needs. In result of the increase of internet usage, there are many data package passed to the network traffic where the data communicate between two-end points (data transmission). This activity must be secure since the user's personal information is a confidential thing. Network has a system to analyse every data passing the traffic and detect a malicious data, called Intrusion Detection System (IDS). IDS needs a detection model to increase its performance in detecting intrusion. The aimed to implement XGBoost algorithm to improve the accuracy score of IDS performance using proposed methods. In this paper, we proposed a detection model using XGBoost algorithm and Sequential Feature Selection (SFS) as feature selection method. These methods have been tested on NSL-KDD dataset. Through the research of implementation of these proposed models, the results obtained by analysing an evaluation metrics such as, accuracy, precision, recall, and f1-score. The result shows the accuracy score reach 99,24%. In the other words, the result is quite high compared to the previous research. Thus, the proposed methods can be used to improve the IDS performance to detect the intrusions and help the network to be more secure. This research still needs development for the further research since the technology grows continuously.

Keywords : *Intrusion detection, Intrusion Detection System (IDS), NSL-KDD dataset Sequential Feature Selection (SFS) XGBoost algorith*

INTRODUCTION

Internet has become an essential need for every human being. In this era, almost every aspect of life depends on internet sources. The role of internet in social life is to support human's activity to be efficient and effective, chiefly as a communication medium (Gani, 2020). Communication turns into the most common activity in internet of things. The more often the communication occurs, the more data transmission activity happens on the internet network. As a result, it causes the internet traffic becomes denser and more varied. In this case, the user information and data security are crucial. Data security is guaranteed in technology development in result of its role to assure the validity and integrity of the data. The data security rate may not to be able to reach secure level fully (Bingham & Garth W. P. Davies, 1978). As the internet technology develops along with the advancement of attack methods in breaking through the network security. The intruders may continuously find a new technology to breach the security tools (Thakkar & Lohiya, 2022). Consequently, the network needs an extra protection from its system that evolve with technology growth.

On the internet, there is a system that has the responsibility to monitor internet traffic and detect data packets that look foreign and dangerous. The system is called the Intrusion Detection System (IDS). IDS works on an internet network with the support of a detection model in carrying out its duties. The detection model can be adopted using machine learning

(ML) algorithms (Ye & Yu, 2015). The use of machine learning algorithms is intended to assist IDS in optimizing its capabilities when detecting network intrusions. Data classification capabilities can be utilized in determining suitability and accuracy values (Guezzaz et al., 2021). The assessment is based on the evaluation metrics generated by the model, especially the accuracy value. The level of accuracy determines how well the model is, implemented in the IDS.

In addition, IDS optimization efforts can also be carried out by maximizing the pre-processing stage. In pre-processing, there is a feature selection stage that has various methods. One method is the wrapper-based method. The wrapper-based method relies on predictive analysis of the learning model used to measure the quality of the features to be selected (Thakkar & Lohiya, 2022). In this study, the wrapper-based method used is forward feature selection, specifically Sequential Feature Selection (SFS). This method selects features automatically based on classifier performance starting from the basic (*null* feature), to produce the best performance (Sharma & Mishra, 2022). Dataset that used in this research is NSL-KDD dataset. This dataset commonly used for network intrusions research. The results of this study obtained evaluation metrics using a classification report and a confusion matrix. Based on this, the accuracy, precision, recall, and f1-score values will be measured. From these results, it will be possible to analyse the use of the XGBoost algorithm as a detection model for IDS in detecting network intrusions.

1 The Proposed Algorithm

1.1 Sequential Feature Selection

In this research, the method to be used is part of the wrapper approaches method. Wrapper approaches use a learner to effectively search for features in a subset of strong influences. One of the methods in the wrapper approach is Sequential Feature Selection (SFS). SFS is a feature selection technique that reduces *n*-dimensional feature space to m-dimensional feature space ($m > n$) based on a greedy search approach. This method selects features automatically based on classifier performance, starting from the basic (null feature) to produce the best performance (Sharma & Mishra, 2022). Aslam et al. (2022) described the SFS performance technique by sequentially adding one feature to the initial feature and providing an assessment of the features without reducing feature criteria. Then, the best features will be selected in one iteration and other features will be added by pairing each feature, so that the algorithm used can achieve maximum performance with the best results. Thus, the possibility of errors can be reduced by

reducing noise and eliminating irrelevant features (Malamatinos et al., 2022). Sequential Feature Selection algorithm is shown in Algorithm 1.

Algorithm 1. Sequential Feature Selection Algorithm	
1.	Create an empty set *(null set)*: $X_n \longrightarrow \{\emptyset\}, n \longleftarrow 0$
2.	Choose optimal feature ina set: $x^+ = argmax_{x+\in X_n}[(X_n + x^+)]$
3.	If $model_{performance(X_n+x^+)} > model_{performance(X_n)}$
	a. Update $X_{n+1} \longleftarrow X_n + x^+$
	b. $n \longrightarrow n + 1$
	c. Repeat Step-2

In wrapper feature selection, the classification method is used as the basis for the classifier, where the classification method is used to evaluate the strength of each feature. In this study, the classifier used is the K-Nearest Neighbor (KNN) classifier. The choice of KNN as a classifier method is due to its simplicity, effectiveness, and intuition in determining features. Although KNN is widely used because of its significant advantages, the KNN algorithm as a classifier has drawbacks. The drawback is in determining the *k* value, which is an important element in processing. This causes the performance results of the KNN classifier to depend heavily on the predetermined *k* value. The *k* value in KNN cannot be ascertained, meaning that there are no numbers that have a definite value to improve classifier performance, so experiments must be carried out using the *k* value to obtain the most optimal results (Wang et al., 2022).

1.2 XGBoost Algorithm

The XGBoost algorithm is a scalable tree boosting algorithm that can be used to solve various problems (Chen & Guestrin, 2016). This algorithm has won several competitions held by Kaggle. Currently, the XGBoost algorithm is widely used in data science. The XGBoost algorithm is a regression and classification algorithm that applies the weak predictor principle and uses decision trees in general (Sugiharti et al., 2021).

Basically, the XGBoost algorithm is a development of the Gradient Boosting algorithm. The difference is in the addition of a regulation term to reduce the risk of overfitting (Liang et al., 2020). The XGBoost algorithm directly adds regular terms and uses the first and second derivative values of the missing functions (Zhang & Gong, 2020). The objective function of the XGBoost algorithm can be stated in a mathematical formula. Adopting the results of research belonging to Luckner et al. (2017), writing the XGBoost

algorithm mathematically can be described in several equations. Initially, the data can be assumed in Equation 1.

$$\mathcal{D} = \{(x_i, y_i) : i = 1 \dots n, x_i \in \mathbb{R}^m, y_i \in \mathbb{R}\} \quad (1)$$

where:

x_i = *x* variable

y_i = *y* variable

n = iterations

m = feature

$\mathbb{R}$ = natural number

Based on the above formula, *n* iterations are carried out with *m* features in each iteration and a corresponding *y* variable. The value $\hat{y}_i$ is defined as the result obtained from a string described in the general model as written in Equation 2.

$$\hat{y}_i = \phi(x_i) = \sum_{k=1}^{K} f_k(x_i) \quad (2)$$

where:

$\hat{y}_i$ = result given by an ensemble

f_k = regression tree

$f_k(x_i)$ = score given by the *k*-th tree to the *i*-th iterations

Then, the stage of minimizing the objective function will be carried out to choose the function f_k can be shown in Equation 3.

$$\mathcal{L}(\phi) = \sum_i l(y_i, \hat{y}_i) + \sum_k \Omega(f_k) \quad (3)$$

where:

l = loss function

Ω = model complexity

The next step aims to reduce the level of model complexity that is too high. The complexity of the model is used to measure how much accuracy the ML model produces in predicting invisible data, so that it can produce good predictions. The complexity of the model must have an optimal value so that overfitting does not occur. This process can be shown in Equation 4.

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda\|w\|^2 \quad (4)$$

where:

γ = parameter of the regulation of the number of leaves

λ = parameter of leaf weight regulation

T = number of leaves

w = magnitude of leaf weights

wherein gamma (γ) and lambda (λ) are parameters to control the excess value (penalty) on the number of leaves (T) and the amount of leaf weight (w). The term "penalty" is a unique feature that XGBoost has, so it can distinguish this method from tree boosting methods in general. The goal is to reduce overfitting and simplify the models generated by the XGBoost algorithm.

An iterative method will be used to minimize the objective function. In the j-th iteration, the function f_j can be added, so that it can be defined in Equation 5.

$$\mathcal{L}^j = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(j-1)} + f_j(x_i)) + \Omega(f_j) \quad (5)$$

By using the Taylor expansion, you can simplify the function and obtain the loss reduction formula after the tree is split from the vertices given in Equation 6.

$$\mathcal{L}_{split} = \frac{1}{2}\left[\frac{(\sum_{i\in\mathcal{I}_L} g_i)^2}{\sum_{i\in\mathcal{I}_L} h_i} + \frac{(\sum_{i\in\mathcal{I}_R} g_i)^2}{\sum_{i\in\mathcal{I}_R} h_i + \lambda} - \frac{(\sum_{i\in\mathcal{I}} g_i)^2}{\sum_{i\in\mathcal{I}} h_i + \lambda}\right] \quad (6)$$

where:

$\mathcal{I}$ = subset of the iterations in the current node

$\mathcal{I}_L$ = subset of the iterations in the left nodes (after the split)

$\mathcal{I}_R$ = subset of the iterations in the right nodes (after the split)

The functions of g_i and h_i are used to find the best split at any given nodes. The formulas are defined as in Equation 7 and 8 bellow.

$$g_i = \partial_{\hat{y}_i^{(j-1)}} l(y_i, \hat{y}_i^{(j-1)}) \quad (7)$$

$$h_i = \partial^2_{\hat{y}_i{}^{(j-1)}} l(y_i, \hat{y}_i{}^{(j-1)}) \quad (8)$$

Luckner et al. (2017) also added information on the optimization of loss function. He mentioned that in regularised the loss function, XGBoost has 2 (two) additional features to prevent overfitting. First, the weight of each new tree can be measured by providing a constant η. This can reduce the impact of a single tree on the final result and make room for subsequent trees, thus improving model performance. Second, using column sampling, this feature is similar to the Random Forest works, where each tree is formed using only column samples from the training dataset.

Based on this description, the XGBoost algorithm was chosen as the ML algorithm to improve the ability of IDS to detect network intrusions.

RESEARCH METHODS

In this study, the XGBoost algorithm, as a detection model, was applied to improve the performance of IDS in detecting intrusions. Meanwhile, the Sequential Feature Selection (SFS) was employed to maximize the data pre-processing, so it can produce the optimal data for the next steps of processing. The flowchart of the research is shown in Figure 1.

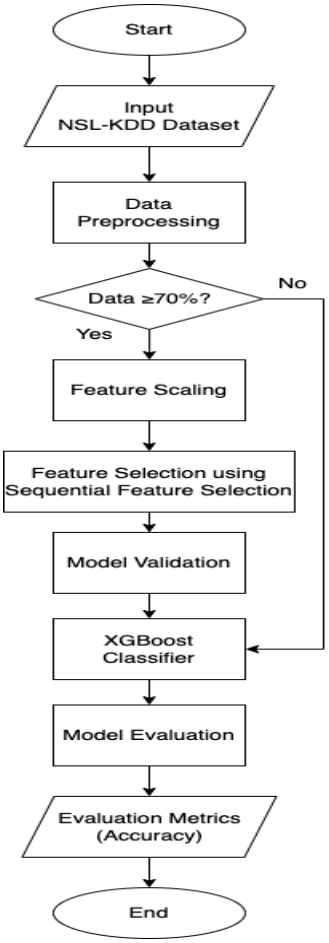

Figure 1. Research Flowchart

Based on the research flowchart, general research steps can be explained. The initial stage of this research is uploading the dataset that will be used. The dataset is wrapped in a comma-separated values (CSV) file type. Then, in the next step, data pre-processing will be carried out, where in this stage a number of processes will be carried out, starting from data cleaning, missing value handling using mean imputation, data analysis, data encoding, data splitting, feature scaling, and feature selection. Data pre-processing is an important step to improve the ability of the ML algorithm model in classification research so as to optimize the classification results. Furthermore, model selection and model validation are used to compare the proposed model with other ML algorithm models. After obtaining the results from the validation model, the next step is to test the data using test data to determine the accuracy, precision, recall, and f1-score values of the classification using the XGBoost algorithm as a proposed detection model. The result is the accuracy level of the proposed model in detecting network intrusions. In the final stage, the results will be analysed by making comparisons based on previous studies.

1.3 Data Collection

The dataset used in this study is NSL-KDD dataset which comes from Kaggle repository. It used kind of supervised data. Basically, The NSL-KDD dataset as a whole consists of 47,737 rows and 42 columns. However, the repository owner, Dhareendra Gupta, was split dataset into 7:3 training and testing data. Therefore, the data used in this processing is the training data, consists of 25.192 rows and 42 columns.

1.4 Exploratory Data Analysis

At this stage, identification of distribution, relationships between data, and invisible patterns will be carried out by looking at the raw data. The relationship between the data in question is to review the relationship between the target and other features.

1.5 Data Pre-processing

Data pre-processing is a crucial stage and is the stage that has the most part in all research phases. Data pre-processing produces reliable and appropriate resources for all data mining algorithms (Shehab et al., 2022).

1.5.1 *Data Cleaning*

The data cleaning process uses the mean imputation method. Mean imputation is a method that handles missing values by substituting them using the arithmetic mean value of other data used (Nikfalazar et al., 2020).

1.5.2 *Data Scaling (Normalization)*

In the data normalization stage, the data will be normalized by scaling in the range 0 to 1. This stage is intended to ensure that the data to be used is of

good quality. So, there is no variable dominates other data variables in a dataset.

1.5.3 *Data Encoding*

So, as to simplify processing, we need encode the dataset since it comes from many data type, such as string and integer. Some algorithm models can only process data in numerical form, while some of the data in the dataset is in the form of strings. The data encoding stage encodes the target class in integer form into numeric, using the numbers 0 and 1.

1.5.4 *Data Splitting*

The data splitting stage is divided into 2 (two) different types of data, namely training data and testing data. The data that will be trained (training data) is 80%, and the data that will be tested using the selected algorithm model (testing data) is 20%. The selection of the data splitting ratio empirically has a good performance on a scale of 70%–80% for training data and 20%–30% for test data. The selection of this 8:2 ratio is based on research done by Gholamy et al. in 2018, which concluded that empirically, the best data splitting ratio is 80% training data and 20% test data. The distribution of the data can be shown in Table 7.

Table 7. Data Splitting

Training Data (%)	Testing Data (%)	Amount of Training Data	Amount of Testing Data	Total
80%	20%	20.153	5.039	25.192

1.6 Feature Selection

This stage serves to select variables to be used in data processing. Feature selection has a very large impact on the performance of the learning model results (Zhong et al., 2020). Selection is made so that data processing can be done more quickly and centrally, resulting in optimal data groups. At this stage, the researcher used the KNN-based Sequential Feature Selection (SFS) method. Effective feature selection can improve model performance and help researchers understand complex data (Shehab et al., 2022). The estimator in SFS utilizes the KNN classifier, so it is necessary to determine the value of *k (n_features)*. This is caused by the influence of the value of *k (n_features)* on processing.

1.7 Model Validation

The model validation stage is the stage of comparing the proposed algorithm model with other classifier algorithm models. In the research that will be conducted, a comparison will be made of the SVM, Logistic Regression, and Gaussian Naïve Bayes algorithms. This stage needs to be done in order to provide a comparison of the performance results of the algorithm that will be used with other algorithms, so that it is more convincing to implement and will produce good performance.

1.8 XGBoost Model

At this stage, classification will be carried out using the XGBoost algorithm. Classification is used to detect attacks on IDS. The XGBoost algorithm has an objective function that can reduce the risk of overfitting so that it can produce an optimal level of accuracy. Trees on XGBoost can create new trees by taking into account the predicted values that have been generated previously to maximize the prediction weight. Based on research by Le et al. (2022), the XGBoost algorithm works by starting from an iterative training process to create a new tree that can resolve the errors and residuals of the previous tree. Then, the process joins the previous tree to produce the final predicted value.

1.9 Accuracy

The output taken in this study is the level of accuracy of the detection of attacks on IDS using the XGBoost algorithm. This can be said to be the output result if the level of accuracy obtained is optimal based on the threshold determined by the researcher, namely > 98%. The accuracy calculation can be done by adding up the true positive and true negative values and dividing it by the total confusion matrix. The confusion matrix table can be seen in Figure 2.

		Predicted Values	
		Negative (0)	*Positive* (1)
Actual Values	*Negative* (0)	*True Negative* (TN)	*False Positive* (FP)
	Positive (1)	*False Negative* (FN)	*True Positive* (TP)

Figure 2. Confusion Matrix

Hence, the accuracy calculation can be written in Equation 9.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TP} \times 100\%$$

where:

TP = True Positive

TN = True Negative

FP = False Positive

FN = False Negative

1.10 Model Evaluation

The model evaluation stage is the stage for assessing model performance. At this stage, the accuracy, precision, recall, and f1-score values can be seen. The model is evaluated using the classification report function provided by scikit-learn.

RESULTS AND DISCUSSION

1.11 Results

This research was conducted to improve the accuracy of the Intrusion Detection System (IDS) using the XGBoost classification algorithm as a detection model. In an effort to optimize the system and process results, this research also utilizes Sequential Feature Selection (SFS) in feature selection.

In this study, data was taken from a dataset that is already available in an open-source data source, namely Kaggle. The dataset used is the NSL-KDD dataset which has become a standard in research related to Intrusion Detection System (IDS). The data consists of 25,192 rows and 42 columns. The data can be seen in Table 8.

Table 8. NSL-KDD Dataset

Index	*duration*	*protocol_type*	*service*	*flag*	*src_bytes*	...	*dst_host_srv_rerror_rate*	*class*
0	0	tcp	ftp_data	SF	491	...	0.00	normal
1	0	udp	other	SF	146	...	0.00	normal
2	0	tcp	private	S0	0	...	0.00	anomaly
3	0	tcp	http	SF	232	...	0.01	normal
4	0	tcp	http	SF	199	...	0.00	normal

The very first step on this research is data cleaning. In a data cleaning process, not only removed some redundant data, but also separated the target label (in this case; class). Then, target label stored in a DataFrame Y. So, we have 2 dataframes, X (the data) and Y (the target label).

Data analysis carried out in this study shows the relationship between the 3 features that are important in determining detection. First, we can take a look at the amount of "class" (Y dataframe) feature as a target. The distribution graphic can be seen in Figure 3.

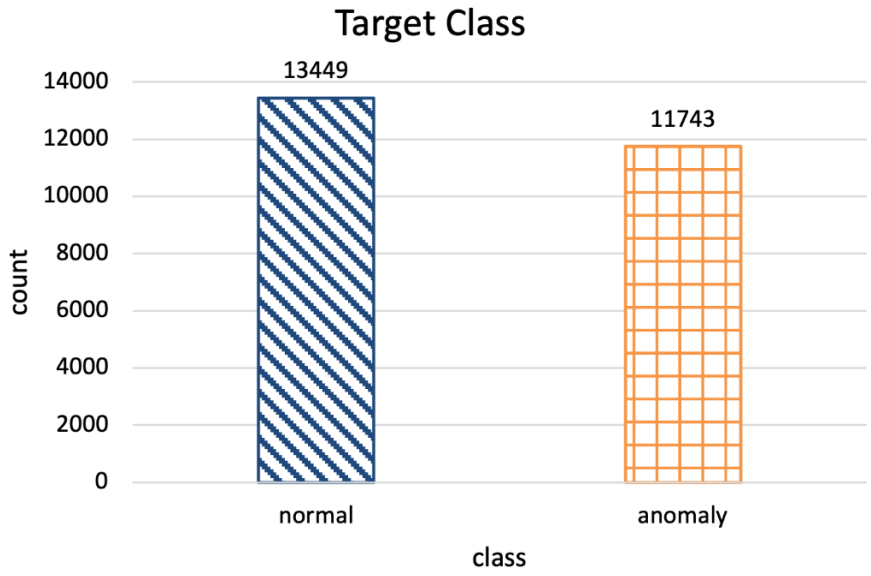

Figure 3. The Distribution between Normal and Anomaly

Based on Figure 3, it can be concluded that of the 25,192 data, there are 13,449 data belonging to the normal "class" and 11,743 data belonging to the anomaly "class". Second, we can analyze the relationship between the "class" attribute as a target and the "protocol_type" attribute, as shown in Figure 4.

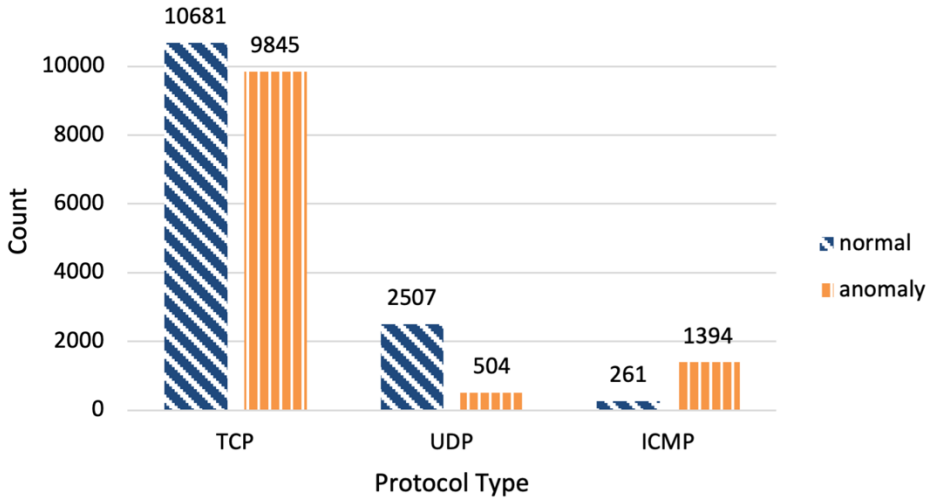

Figure 4. Class Distribution on Traffic Data

Based on Figure 4, most classes in TCP traffic are normal. However, the comparison with normal classes with anomalies is not significant or much different. Likewise, with UDP traffic. Whereas in ICMP traffic, the majority of data is in the anomaly class. Lastly,

we can analyze the amount of data that is in the normal and anomalous classes in the "flag" feature, as can be seen in Figure 5.

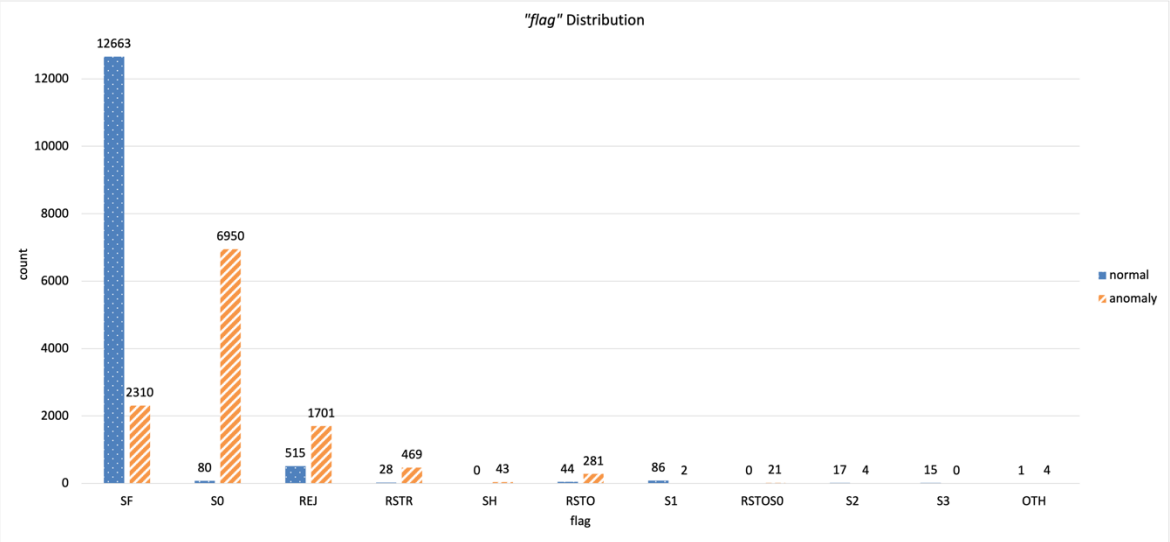

Figure 5. Traffic Distribution on the Basis of Flags

From the graph above, the traffic distribution based on flags was also uneven where most of it had SF (Sign Flag). Most of the traffic with SF was normal, while that had S0 flag had anomaly.

After analysing the data, the necessary pre-processing stage is carried out to optimize model performance in measuring the accuracy of IDS in detecting network intrusions. data pre-processing intended to perform data scaling. This is because there are still data points that have significant differences. Besides that, there is also a data splitting process. In the data splitting stage, the two data frames are split into training data and test data. The splitting used 8:2 ratio, with 80% for training data and 20% for testing data. Afterwards, we can start training the data using the proposed method. At first, we selected the best feature for processing using a feature selection method, in this case is SFS. We used KNN as SFS estimator. This estimator will train the data to produce the best feature that the training data has. The results show the amount of feature that we will use for the model validation.

In the model validation, we will compare our method and 3 others machine learning algorithm, such as Support Vector Machine (SVM), Logistic Regression, and also Gaussian Naïve Bayes. This model validation is carried out the comparison of each model's performance. The comparison graph can be seen on Figure 6.

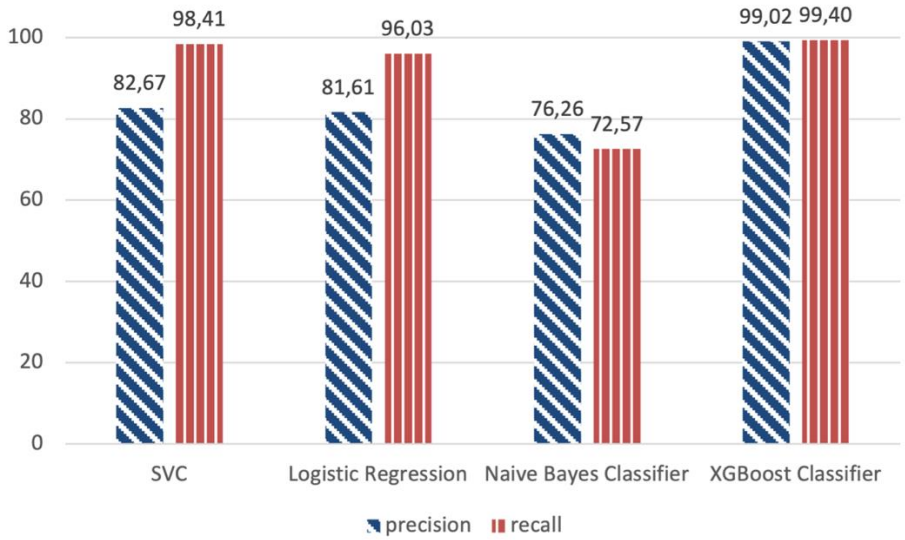

Figure 6. The Model Validation Results

As on Figure 6 that the proposed method has the higher results with the precision score is 99,02% and recall is 99,40%. It means that we can use this proposed model to reach our goal in implementing XGBoost as detection model in an IDS to detect network intrusions.

After selecting the best model based on the model evaluation, the next step is to test the model using test data. Then, the result is a model evaluation using a classification report and confusion matrix. The classification can be shown on Figure 7.

```
**************** XGBoost Classifier Model Testing ***************
[[2339   26]
 [  12 2662]]
----------------
              precision    recall  f1-score   support

      normal       0.99      0.99      0.99      2365
     anomaly       0.99      1.00      0.99      2674

    accuracy                           0.99      5039
   macro avg       0.99      0.99      0.99      5039
weighted avg       0.99      0.99      0.99      5039
```

Figure 7. XGBoost Confusion Matrix and Classification Report

Based on Figure 7, we can calculate the accuracy score using the accuracy formula based on Equation 9. The calculation can be written on Equations below.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \times 100$$

$$Accuracy = \frac{2.662 + 2.339}{2.662 + 26 + 12 + 2.339} \times 100$$

$$= \mathbf{99,24}\%$$

From the calculation, we can get the accuracy score of the XGBoost algorithm is 99,24%. The score is quite high compared to the previous research. By that means, the XGBoost algorithm can be used for detecting network intrusions in an IDS.

1.12 Discussion

The research that has been done is research on the use of the XGBoost model on the Intrusion Detection System (IDS) in detecting intrusion networks. In this study, the dataset used is the NSL-KDD dataset. When uploading a dataset for processing, the data entered is training data. In other words, this dataset has undergone the data splitting process. Based on Dhareendra Gupta's Kaggle, which is a reference source for researchers conducting this research, dividing the dataset at the beginning of the warehouse is to prevent data leakage. The data split by Dhareendra Gupta uses a 7:3 ratio, with 70% training data and 30% test data.At the pre-processing stage, there is data splitting which is intended to shorten processing time (running time) and is also more efficient for models to conduct learning and testing of data. However, the comparison ratio used in the pre-processing stage is 8:2, with 80% training data and 20% testing data.

Based on model validation result, it can be concluded that the use of the XGBoost algorithm on training data is the best choice by looking at the results of the validation model, compared to the other methods. The XGBoost algorithm gets a mean precision value of 99.02% and a main recall of 99.40% compared to other models. For this reason, the XGBoost algorithm is used as the detection model algorithm. In the final results of the confusion matrix and the large value of accuracy in predicting, it states that the XGBoost algorithm is capable of detecting network intrusions with an accuracy rate of 99.24%.

With the help of the detection model in the form of the XGBoost algorithm, IDS as a detection system can become an optimal security system, so that it will properly maintain the security of residential user data. Besides that, in the process, this research utilizes Sequential Feature Selection (SFS) as a feature selection model, which can also support the performance of the classifier model. KNN-based SFS feature selection works by selecting the best features based on the cross-validation score in an estimator. The KNeighborsClassifier() classifier has a default k value of 5. The choice of the k=5 value is based on experiments conducted by Zhao et al. in 2022. Zhao et al. used k = 5 and got the best result in their research, which was 95.02%. In addition, this selection is also based on the experiments that have been done. When given a value of k = 3, the resulting accuracy level is 0.9902 or 99.02%. Meanwhile, when the trial

was carried out at the default setting (k = 5), the results showed an increased level of accuracy from the previous experiment, which was 0.9924 or 99.24%. Therefore, the value of k = 5 was chosen to support the ability of the classifier algorithm to obtain optimal results.

CONCLUSION

Based on the research, it can be concluded that the implementation of XGBoost algorithm for an IDS can fit the best to detect network intrusions. Classification measurements are based on accuracy, precision, recall, and f1-score values. The accuracy results obtained from the use of the XGBoost algorithm in the study were 99.24%, 99.02% precision, 99.40% recall, and 0.99 f1-score. In that sense, XGBoost's level of accuracy in detecting network intrusions on IDS can be said to be optimal. These results have been compared with those of previous studies. The results of the research that has been done show an increase. However, when compared to the Kaggle source used, there is a decrease of 0.56%. This is because the process of feature selection uses a different method.

REFERENCES

Aslam, N., Khan, I. U., Mirza, S., Alowayed, A., Anis, F. M., Aljuaid, R. M., & Baageel, R. (2022). Interpretable Machine Learning Models for Malicious Domains Detection Using Explainable Artificial Intelligence (XAI). *Sustainability (Switzerland)*, *14*(12). https://doi.org/10.3390/su14127375

Bingham, J. E., & Garth W. P. Davies. (1978). *A Handbook of Systems Analysis*. Macmillan International Higher Education.

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, *13-17-August-2016*, 785–794. https://doi.org/10.1145/2939672.2939785

Gani, A. G. (2020). Sejarah dan Perkembangan Internet di Indonesia. *Jurnal Mitra Manajemen*, *5*(2).

Gholamy, A., Kreinovich, V., & Kosheleva, O. (2018). A Pedagogical Explanation A Pedagogical Explanation Part of the Computer Sciences Commons. *Departmental Technical Reports (CS)*. https://scholarworks.utep.edu/cs_techrephttps://scholarworks.utep.edu/cs_techrep/1209

Luckner, M., Topolski, B., & Magdalena Mazurek. (2017). Application of XGBoost Algorithm in Fingerprinting Localisation Task. *Computer Information Systems and Industrial Management (CISIM)*, *10244*, 661–671. https://doi.org/10.1007/978-3-319-59105-6

Malamatinos, M. C., Vrochidou, E., & Papakostas, G. A. (2022). On Predicting Soccer Outcomes in the Greek League Using Machine Learning. *Computers*, *11*(9). https://doi.org/10.3390/computers11090133

Nikfalazar, S., Yeh, C. H., Bedingfield, S., & Khorshidi, H. A. (2020). Missing Data Imputation using Decision Trees and Fuzzy Clustering with Iterative Learning. *Knowledge and Information Systems*, *62*(6), 2419–2437. https://doi.org/10.1007/s10115-019-01427-1

Sharma, A., & Mishra, P. K. (2022). Performance Analysis of Machine Learning based Optimized Feature Selection Approaches for Breast Cancer Diagnosis. *International Journal of Information Technology (Singapore)*, *14*(4), 1949–1960. https://doi.org/10.1007/s41870-021-00671-5

Shehab, N., Badawy, M., & Ali, H. A. (2022). Toward Feature Selection in Big Data Preprocessing based on Hybrid Cloud-Based Model. *Journal of Supercomputing*, *78*(3), 3226–3265. https://doi.org/10.1007/s11227-021-03970-7

Sugiharti, E., Arifudin, R., Wiyanti, D. T., & Susilo, A. B. (2021). Convolutional Neural Network-XGBoost for Accuracy Enhancement of Breast Cancer Detection. *Journal of Physics: Conference Series*, *1918*(4). https://doi.org/10.1088/1742-6596/1918/4/042016

Thakkar, A., & Lohiya, R. (2022). A Survey on Intrusion Detection System: feature selection, model, performance measures, application perspective, challenges, and future research directions. *Artificial Intelligence Review*, *55*(1), 453–563. https://doi.org/10.1007/s10462-021-10037-9

Wang, Y., Pan, Z., & Dong, J. (2022). A New Two-layer Nearest Neighbor Selection Method for kNN Classifier. *Knowledge-Based Systems*, *235*, 107604. https://doi.org/10.1016/J.KNOSYS.2021.107604

Zhong, W., Yu, N., & Ai, C. (2020). Applying Big Data Based Deep Learning System to Intrusion Detection. *Big Data Mining and Analytics*, *3*(3), 181–195. https://doi.org/10.26599/BDMA.2020.9020003