MORFAI JOURNAL
Multidisciplinary Output Research For Actual and International Issue

Multidisciplinary Output Research For Actual and International Issue (MORFAI Journal) ISSN (e): 2808-6635
Volumes 6 No. 2 (2026)

DECISION TREE BASED INTERNET SIGNAL QUALITY ANALYSIS ON TELKOM INFRASTRUCTURE

Yeni Fitri Afidah[1], Sri Mujiyono[2]
Informatics Engineering, Universitas Ngudi Waluyo, Semarang, Indonesia
E-mail: 94.yenifitria@gmail.com[1], mujiyn80@gmail.com[2]

Received	: 20 November 2025	Published	: 08 January 2026
Revised	: 01 December 2025	DOI	: https://doi.org/10.54443/morfai.v6i2.4949
Accepted	: 25 December 2025	Publish Link	: https://radjapublika.com/index.php/MORFAI/article/view/4949

Abstract

This study analyzes internet signal quality on Telkom infrastructure using the Decision Tree algorithm. Utilizing 5,000 data points from Kaggle, the research classifies network quality into three categories: Good, Fair, and Poor based on parameters such as download speed, ping latency, and packet loss. The evaluation results show that the Decision Tree model achieved an accuracy rate of 98%. Parameters such as download speed and ping latency were identified as the most dominant factors in determining signal quality. These findings prove that a machine learning approach is effective in generating easily interpretable decision rules for network service optimization.

Keywords: Decision Tree, Signal Quality, Telkom, Network Accuracy.

INTRODUCTION

The development of information and communication technology has made the internet a major need in various sectors, such as education, business, government and public services. The need for reliable internet services continues to increase as digital applications grow; therefore, good quality internet signals are essential to maintain Quality of Service (QoS) and optimal user experience. Common problems such as high latency, packet loss, and throughput instability can reduce service quality and impact overall user satisfaction (Salau & Beyene, 2024). As the main telecommunications service provider in Indonesia, Telkom has a responsibility to guarantee good network performance for its customers. Network quality evaluation generally involves analysis of QoS parameters such as latency, throughput, and packet loss. Traditional methods such as packet inspection or port-based approaches are often less effective in dealing with modern traffic variability and encrypted data (Salau & Beyene, 2024). To overcome these limitations, the approach is based on machine learning has become an effective alternative method for network data classification. In context network traffic classification, some studies show that algorithms supervised learning such as Decision Tree, Random Forest, and k-Nearest Neighbors (KNN) can provide good performance in traffic classification compared to traditional methods (Lohiya & Bamnote, 2025).

Several previous studies have discussed internet network quality analysis using a Quality of Service (QoS) based approach and classification methods. Research by Al-Fuqaha et al. (2015) shows that QoS parameters such as latency, throughput, and packet loss have a significant influence on network service quality (Al-Fuqaha et al., 2015). Meanwhile, Chen et al. (2020) stated that the application of machine learning methods can increase the accuracy of network performance analysis compared to conventional methods. Nonetheless, part of the research still produces models that are difficult for network practitioners to interpret, so a classification approach is needed that is not only accurate but also easy to understand. Recent studies have also shown the relevance of using the Decision Tree method in the context of QoS and network classification. For example, Decision Tree has been evaluated alongside other algorithms for internet traffic classification, and in general these algorithms demonstrate the competitive ability to classify network traffic based on QoS features (Lohiya & Bamnote, 2025). Additionally, integration machine learning like the Decision Tree in software-defined networking (SDN) it has been shown to improve traffic classification accuracy as well as QoS management capabilities in complex networks (Salau & Beyene, 2024). The main advantage of using the Decision Tree in this research is its ability to produce decision rules (if–then rules) which is explicit and easy to interpret, in contrast to several other algorithms such as Naive Bayes or KNN which, although accurate, tend to produce models that are less easy to read. This is important because these decision rules can be used directly by Telkom network technicians for operational analysis and network quality improvements in the field. Based on in this aspect, this study used the Decision Tree algorithm to classify internet signal quality based on QoS data from the Kaggle public dataset. The numerical target internet speed is then transformed into three quality categories: Ok, Enough, and Bad, so that it can be analyzed in classification. The results of the study are expected to provide not only signal quality predictions, but also rules that can be used as technical guidance for network practitioners.

METHOD

1. Types and Approaches of Research

This research uses a quantitative approach using data mining and machine learning methods. This approach aims to classify internet signal quality based on network parameters using the Decision Tree algorithm.

2. Research Dataset

The datasets used in this study are Internet Speed Prediction Dataset obtained from the Kaggle.Dataset platform, it is read using the Python programming language with the Pandas library to facilitate the analysis and data processing process. The dataset consists of 5,000 data with various network parameters that affect internet performance, such as ping latency, download speed, upload speed, packet loss, signal strength, network congestion level, and ISP quality.

3. Research Stages

The stages of research carried out are:

a) Data Collection

The following code is used to load a quality dataset of internet signals that have been stored in CSV format.

```
import pandas as pd
df = pd.read_csv("Dataset_Quality_Signal.csv")
print(df)
```

Datasets are read and managed using the Python programming language with the Pandas library to facilitate the data analysis and pre-processing process. The datasets used in this study are Internet Speed Prediction Dataset obtained from the Kaggle platform. This dataset consists of 5,000 data with various network parameters that affect internet performance, such as ping latency, download speed, upload speed, packet loss, signal strength, network congestion level, and ISP quality. Based on descriptive statistical analysis, internet speed values on the dataset range from 80.61 Mbps to 3364.87 Mbps, with a median of 1018.98 Mbps. This shows that the dataset has a fairly wide distribution of internet speeds.

b) Data Pre-Processing

The following code is used to detect empty values (missing values) and delete incomplete data.

```
df.isnull().sum()
df = df.dropna()
```

The data pre-processing stage is carried out by checking the presence of empty values in dataset. Data that contains blank values removed to ensure the quality of data used in the model training and testing process. Next, the numerical Internet_speed variable target is transformed into a categorical variable for classification purposes.

Target transformation is carried out using a quantile (percentile) approach so that category division is objective and balanced. The percentile limits used are:

a) 25% percentile (Q1) = 430.81 Mbps
b) 75% percentile (Q3) = 1951.30 Mbps

Based on these values, the signal quality categories are determined as follows:

Internet Speed Range (Mbps)	Quality Category
< 430,81	Bad
430.81 – <1951.30	Enough
≥ 1951.30	Ok

This approach results in a more balanced class distribution and allows optimal application of classification algorithms.

c) Feature Selection

The selected attribute is a network parameter relevant to the quality of the internet signal.

```
X = df[
   ["Ping_latency", "Download_speed", "Upload_speed",
    "Packet_loss_rate", "Signal_strength",
    "Network_congestion", "ISP_quality"]
]
```

Input variables used in this study included ping latency, download speed, upload speed, packet loss rate, signal strength, network congestion rate, and ISP quality. These variables were chosen because they have a direct influence on the quality of internet services.

d) Variable Target Transformation

Internet_speed numerical targets are transformed into signal quality categories using a quantile approach.

```
q25 = 430.808067
q75 = 1951.295923

def category_quality(speed):
    if speed >= q75:
        return "Good"
  elif speed >= q25:
        return "Enough"
    else:
        return "Bad"

df["Quality_category"] = df["Internet_speed"].apply(quality_category)
print(df["Quality_Category"].value_counts())
```

The numerical Internet_speed variable target is transformed into a categorical variable using a percentile (quantile) approach. 25% percentile limit amounting to 430.81 Mbps and a 75% percentile of 1951.30 Mbps were used to form three signal quality categories, namely Bad, Enough, and Good. This approach aims to produce a balanced class distribution.

e) Division

The dataset is divided into training data and test data in a ratio of 70:30 stratified.

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(
    X, y,
 test_size=0.3,
 random_state=42,
    stratify=y
)
```

At this stage, the dataset that has gone through a pre-processing process is divided into two parts, namely training data (training data) and test data (testing data). Of the total 5,000 data, 70% was used as training data and 30% as test data. Data sharing is carried out using the Stratified Sampling method, which is a data sharing technique that maintains the proportion of each target class in training data and test data. With this approach, the distribution of signal quality categories Ok, Enough, and Bad the training data and test data remain balanced and representative of the entire dataset. The use of stratified sampling is important to avoid model bias towards certain classes and ensure that the resulting model performance truly reflects classification capabilities across all signal quality categories.

f) Development

Model The classification model was constructed using the Decision Tree Classifier algorithm with criteria for selecting Entropy attributes. The entropy criterion is used to measure the level of uncertainty of the data and determine the attribute with the highest information gain as the dividing node in the decision tree. In this study, the max_depth parameter was limited to 4 levels. Decision tree depth restrictions are made to prevent overfitting, which is a condition when the model is too complex and only memorizes training data without being able to generalize patterns to new data. By limiting tree depth, models are expected to be able to capture key patterns from internet network data while producing tree structures that are simpler and easier to interpret. Here is the python code for the model.

```
model = DecisionTreeClassifier(
    criterion="entropy",
 max_depth=4,
 random_state=42
)
model.fit(X_train, y_train)
```

g) Evaluation

Evaluation is carried out using confusion matrix, accuracy and classification report. Evaluation of model performance was performed using Confusion Matrix and Accuracy Score. The confusion matrix is used to analyze the agreement between the actual class and the predicted result class in each signal quality category, while the accuracy score is used to measure the overall level of accuracy of the classification. Results evaluation shows that the Decision Tree model is able to achieve an accuracy level of 98%, which indicates excellent classification performance. Nevertheless, there is still a small number of misclassifications. This error generally occurs in data that has network parameter values around threshold limits (thresholds) between categories, especially between categories Enough and Ok or between Enough and Bad. This shows that the data characteristics in the category switching area are sufficiently similar to be difficult for the model to distinguish expressly. Here is the python code for confusion matrix.

```
y_pred = model.predict(X_test)

cm = confusion_matrix(y_test, y_pred, labels=["Good","Enough","Bad"])
print(cm)
print(accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

4. Decision Tree Algorithm

Decision Tree is a classification algorithm that works by forming a decision tree structure based on the data gain value of each attribute. The attribute with the highest information gain will be selected as the separator node, so that the data can be separated optimally into target classes (Jijo & Abdulazeez, 2021). The main advantage of the Decision Tree algorithm in the context of Telkom's infrastructure is its ability to produce decision rules (if–then rules) which is easy for humans to understand. This rule is very useful for network technicians because it can be used directly as a guide in analyzing and dealing with signal quality problems in the field. The Decision Tree algorithm has been widely used in various classification studies because of its ability to form models that are interpretable and easy to understand. Quinlan (1993) introduced the concept of a Decision Tree based on information gain as the basis for forming a decision tree structure, while Breiman et al. (1984) showed that Decision Trees have competitive performance in classification problems with varying data complexity.

5. Evaluation Method

The model evaluation method was performed using Confusion Matrix, Precision, Recall, and F1-Score for each signal quality category.

- Precision shows the degree of precision of the model's predictions, that is, how much of the proportion of data predicted in a class actually belongs to that class.
- Recall shows the model's ability to find all the correct data in a class.
- F1-Score is a harmonization value between precision and recall, which is used to assess the balance of model performance.

The use of these three metrics is important to provide a more comprehensive picture of model performance, especially in multiclass data conditions such as internet signal quality classification.

The use of precision, recall and F1-score metrics is very important in multiclass classification problems because it is able to provide a more balanced picture of model performance compared to accuracy alone. According to Powers (2011), F1-score is a metric evaluation which is effective for assessing the balance of performance of classification models, especially when there is a class imbalance.

RESULTS AND DISCUSSION

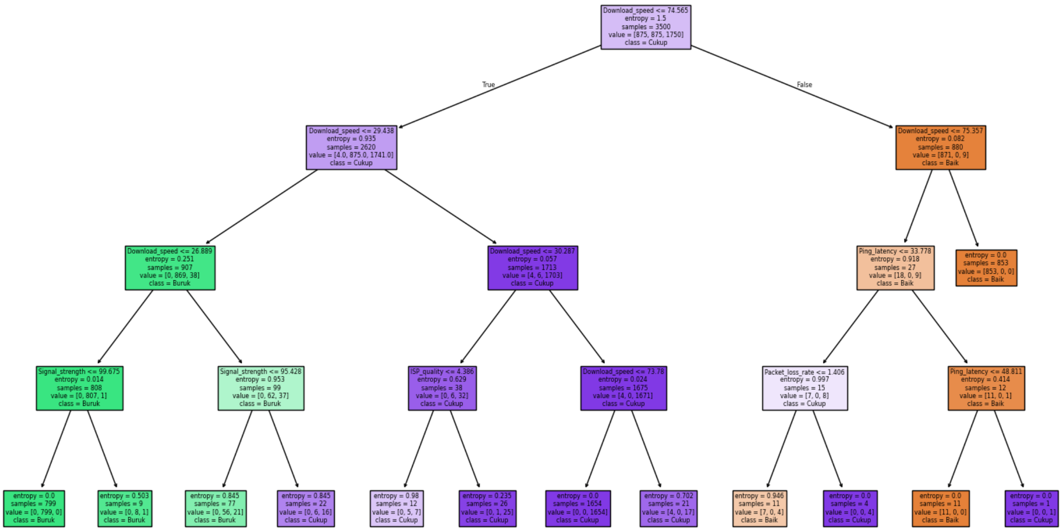
a) Data Pre-processing Results

The public dataset from Kaggle used has gone through a pre-processing phase, including data cleanup and variable target transformation. Displays the top 10 data from the dataset that has gone through stages Feature Selection.

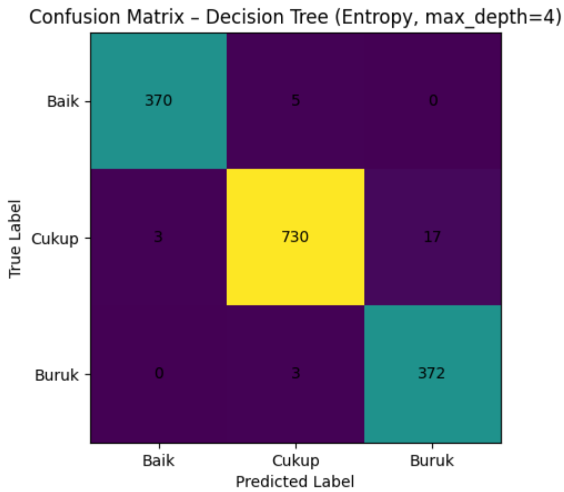
```
=== 10 DATA TERATAS ===
   Ping_latency  Download_speed  Upload_speed  Packet_loss_rate  \
0     21.854305       42.395374     19.934759          0.999340
1     47.782144       49.976388     17.979781          1.493494
2     37.939727       86.182002     10.455388          1.125334
3     31.939632       37.300417     31.148800          0.166605
4     12.020839       87.616720     24.877960          0.371160
5     12.019753       13.372771     43.553648          0.438656
6      7.613763       78.795852      3.541260          0.515729
7     43.977927       85.517025     32.905661          1.130807
8     32.050176       22.272677     38.621546          1.007932
9     36.863266       45.882921     38.455355          0.406320
```

b) Signal Quality Classification Results

Based on classification report, category Bad has a precision value of 0.96, which is slightly lower than other categories. This indicates that a small portion of the data is predicted to be Bad actually has characteristics that are close to categories Enough. This phenomenon indicates the similarity of network parameter patterns between signal conditions Bad and Enough, especially in data that are around the classification threshold. However, the recall value is high in the category Bad shows that the fixed model is able to accurately recognize almost all data with poor signal quality. Overall, these results prove that the Decision Tree algorithm is not only able to provide high accuracy, but also produces decision rules that can be used as a basis for technical analysis to improve the quality of internet network services. The classification tree can be seen in the image below.



c) Model Evaluation Results



1. Good Category

A total of 370 data with signal quality Ok successfully classified as Ok. There are 5 data that are incorrectly classified as Enough. There is no data Ok which is incorrectly classified as Bad.

2. Category Enough

A total of 730 data were correctly classified as Enough. There are 3 data that are incorrectly classified as Ok. A total of 17 data were misclassified as Bad.

3. Bad Category

A total of 372 data were correctly classified as Bad. There are 3 data that are incorrectly classified as Enough. There is no data Bad which is incorrectly classified as Ok.

d) Accuracy, Precision and Recal Analysis Results

The classification report in this image presents precision, recall, f1-score and support values for each internet signal quality category, namely Good, Enough and Bad.

	precision	recall	f1-score	support
Baik	0.99	0.99	0.99	375
Buruk	0.96	0.99	0.97	375
Cukup	0.99	0.97	0.98	750
accuracy			0.98	1500
macro avg	0.98	0.98	0.98	1500
weighted avg	0.98	0.98	0.98	1500

1. Good Category
    - Precision:0.99
      Shows that 99% of the predicted data is quality Ok really falls into categories Ok.
    - Recall:0.99
      Shows that 99% of data Ok successfully recognized correctly by the model.
    - F1-score:0.99
      Value this shows an excellent balance between precision and recall.
    - Support: 375 data

   These results show that the model is very accurate in identifying the quality of category signals Ok.
2. Bad Category
    - Precision:0.96
      That is, 96% prediction Bad according to actual conditions.
    - Recall:0.99
      Shows that almost all data Bad successfully classified correctly.
    - F1-score: 0.97
    - Support: 375 data

   A precision value that is slightly lower than other categories indicates the presence of a small portion of data Bad which has characteristics similar to categories Enough.
3. Category Enough
    - Precision: 0.99
    - Recall: 0.97
    - F1-score: 0.98
    - Support: 750 data

   Category Enough has the largest amount of data. A slightly lower recall value indicates that a small portion of the data Enough classified into other categories, in particular Bad, which corresponds to the confusion matrix results.

CONCLUSION

Several previous studies have shown that the Decision Tree algorithm can be used to classify network activity data with high accuracy and produce decision rules that are easy to interpret.(Sitanggang & Private, 2025). Based on the research results, it can be concluded that network parameters such as ping latency, download speed, upload speed, packet loss rate, and signal strength can be used for analyze internet signal quality. The transformation of internet speed targets into quality categories allows the application of the Decision Tree algorithm as a classification method.

REFERENCES

Cisco Systems, Inc. (2023). Cisco annual internet report (2018–2023). Cisco. https://www.cisco.com/site/us/en/solutions/annual-internet-report/index.html

Han, J., Kamber, M., & Pei, J. (2012). Data mining: Concepts and techniques (3rd ed.). Morgan Kaufmann. https://shop.elsevier.com/books/data-mining/han/978-0-12-811760-6

International Telecommunication Union. (2016). ITU-T Recommendation Y.1541: Network performance objectives for IP-based services. ITU-T. https://www.itu.int/rec/T-REC-Y.1541/en

Kaggle. (2023). Internet speed prediction dataset. https://www.kaggle.com/datasets/getanmolgupta01/internet-speed

Katz, R. L., & Koutroumpis, P. (2013). Measuring digitization: A growth and welfare multiplier. Technovation, 33(10–11), 314–319. https://www.sciencedirect.com/science/article/abs/pii/S0166497213000667

Telkom Indonesia. (2023). Annual report of PT Telekomunikasi Indonesia Tbk. https://www.telkom.co.id/sites/hubungan-investor/id_ID/page/laporan-1025

Lohiya, P. B., & Bamnote, G. R. (2025). Internet Traffic Classification through Supervised Learning: Exploring Machine Learning Techniques. *Intelligent Methods in Engineering Sciences*, 4(1), 8–14. Retrieved from https://doi.org/10.58190/imiens.2025.119

Salau, A. O., & Beyene, M. M. (2024). Software defined networking based network traffic classification using machine learning techniques. *Scientific Reports*, 14, 20060. https://doi.org/10.1038/s41598-024-70983-6

Nafis, Z. M. J., Nazilla, R., Nugraha, R., & ’Uyun, S. (2024). Comparison Algorithm Decision Tree and K-Nearest Neighbor for Classification Attack Network IoT. *Computics: Journal System Computer, 13*(2), 245–252. https://doi.org/10.34010/computika.v13i2.12609

Rosyidin, Z. U., Muladi, M., & Handayani, A. N. (2025). Determining Quality of Service (QoS) of End-User Internet Networks with Data Sniffing and Classification Algorithms. *International Journal of Artificial Intelligence Research, 9*(1). https://doi.org/10.29099/ijair.v9i1.1444

Muhajriyah, S. J., Ruslan, & Firdaus. (2024). Analysis and Classification Of Wireless LAN Network Service Quality Using The Naive Bayes Algorithm. *Journal of Computers, Information and Technology, 4*(2). https://doi.org/10.53697/jkomitek.v4i2.1841

Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015).

Internet of Things: A survey on enabling technologies, protocols, and applications. IEEE Communications Surveys & Tutorials, 17(4), 2347–2376. https://doi.org/10.1109/COMST.2015.2444095

Chen, M., Challita, U., Saad, W., Yin, C., & Debbah, M. (2020). Artificial neural networks-based machine learning for wireless networks: A tutorial. IEEE Communications Surveys & Tutorials, 21(4), 3039–3071. https://doi.org/10.1109/COMST.2019.2926625

Sitanggang, N. S. M., Personal, O., & Hidayat, H. (2024). Implementation of the decision tree method for detecting attacks in networks. Journal of Artificial Intelligence and Engineering Applications, 4(1), 45–52. https://journal.ioinformatic.org/index.php/JAIEA/article/view/1272

Charbuty, B., & Abdulazeez, A. (2021). Classification based on decision tree algorithm for machine learning. Journal of Applied Science and Technology Trends, 2(1), 20–28. https://doi.org/10.38094/jastt20165