Eduvest – Journal of Universal Studies
Volume 5 Number 1, January, 2025
p- ISSN 2775-3735- e-ISSN 2775-3727

Classification of Health Index of Distribution Substations using Supervised Learning Analysis with SVM Method

Donny Zaviar Rizky, Suprijadi
Institut Teknologi Bandung, Indonesia
Email: donny.zaviar@gmail.com, suprijadi@itb.ac.id

ABSTRACT

As the only electricity provider in Indonesia, PLN is required to be reliable in distributing electrical energy to customers, this is greatly influenced by several PLN assets in the form of distribution substations. The function of this distribution substation is quite crucial in carrying out PLN's business processes to distribute electrical energy. In this study, efforts were made to improve the reliability of distribution substations by knowing the health index in accordance with EDIR PLN No. 017 concerning Distribution Transformer Maintenance Methods Based on Asset Management Principles as the Basis of the Health Index. By knowing the health level of the transformer at the distribution substation, the substation that has substandard criteria can be prioritized for maintenance. The research carried out was to take a sample in 1 month, namely March 2024, from a total of 239 substations, which were then classified using the Support Vector Machine (SVM) method which was compiled in the Python programming language which had been labeled with criteria on each substation. The criteria used in accordance with the PLN EDIR No. 017 PLN are Good, Sufficient, Less and Poor. By using Machine Learning according to the Support Vector Machine (SVM) method with Supervised Learning, after the data samples were labeled, then from 239 sample data, it was divided into 2 data, namely training data and test data. In this study, the experiment was carried out with changes in the training data by 60%, 70%, 80% and 90% which were then evaluated for accuracy using libary from Python.

KEYWORDS *Distribution Substation, Support Vector Machine (SVM), Support Vector Machine, Pyhton, Machine Learning*

INTRODUCTION

In line with the PT PLN (Persero) program, namely Transformation, one of the most influential targets to improve the reliability of electricity distribution is the distribution system. A reliable and efficient electricity distribution system is an important component in maintaining the quality of energy services to consumers. PT PLN (Persero).

One way to improve the reliability of this distribution system is by increasing the reliability of distribution substation assets. With the condition of substation

How to cite: Donny Zaviar Rizky et al. (2025). Classification of Health Index of Distribution Substations using Supervised Learning Analysis with SVM Method. Journal Eduvest. 5(1): 1262-1272
E-ISSN: 2775-3727

assets that are quite old, the level of disturbance in the transformer is also high. This is inversely proportional to the financial condition used for the asset maintenance budget. In conditions like this, PT PLN (Persero) is required to carry out effective maintenance on its assets. Therefore, this study is expected to help provide recommendations for prioritizing maintenance on assets that have substandard criteria. Of course, the research conducted is based on regulations and policies from PT PLN (Persero), namely EDIR PLN No. 017 which in this policy regulates how distribution substation assets, especially transformers, are given 4 classes of criteria to be Good, Adequate, Less and Bad. Each of these criteria has special characteristics in the samples taken, including *Visual Inspection, Load Reading and Profiling* and *Oil Quality Analysis*.

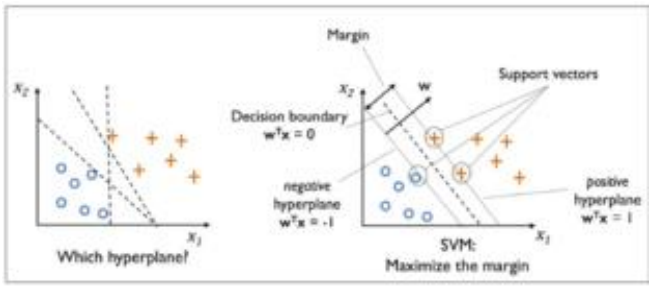
To overcome this problem, a sampling study was carried out to meet the needs of the appropriate PLN EDIR No. 017. This sampling was carried out in March 2024 at UP3 Ciracas and successfully sampled a total of 239 data. This data processing uses one of the methods of Machine Learning in the form of Supervised Learning, namely the Support Vector Machine (SVM). With this method modeling, it is hoped that it can solve the case in the classification of the distribution substation which has various classes of requirements. Combined with the Python programming language, it is hoped that the program to be built can produce a model that has a fairly high level of accuracy. Python has the advantage of being able to solve model problems with various parameters in a fairly short time, for this reason it is the basis for choosing this programming language to use. Python also has libraries that are quite helpful in solving similar cases. From the model that has been built using the SVM method in Python, it is designed to provide an output in the form of transformer health condition criteria. From the results of this output, it is a reference for asset maintenance.

This classification model is expected to help provide the recommendations needed by PLN in order to prioritize the maintenance of existing assets with limited budget conditions with criteria that have met the policies of PLN.

Theoretical foundations

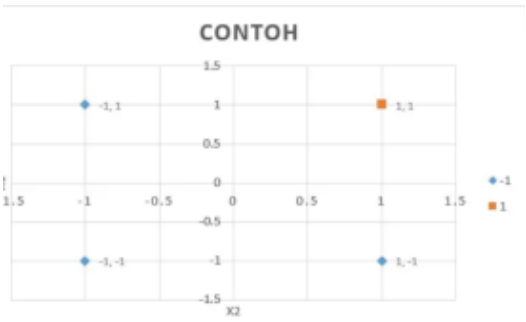
Support Vector Machine (SVM) is one of the methods in supervised learning that is usually used for classification (such as Support Vector Classification) and regression (Support Vector Regression). In classification modeling, SVM has a more mature and mathematically clearer concept compared to other classification techniques. SVM can also overcome classification and regression problems with linear and non-linear. SVM is used to find the best hyperplane by maximizing the distance between classes. Hyperplane is a function that can be used to separate classes. In 2-D the function used for classification between classes is called line whereas, the function used for class-wide classification in 3-D is called plane

similarly, while the function used for classification within higher dimensional classrooms is called hyperplane.



The hyperplane found by SVM is illustrated as shown in the image above in the middle between two classes, meaning that the distance between the hyperplane and the data objects is different from the adjacent (outermost) class which is marked with blank and positive circles. In SVMs, the outermost data object closest to the hyperplane is called a support vector. Objects called support vectors are the most difficult to classify because they are almost overlapping with other classes. Given its critical nature, only this support vector is taken into account to find the most optimal hyperplane by SVM. From the example above, a plot of sample data is obtained as described in the figure below.

Y	X1	X2
1	1	1
-1	1	-1
-1	-1	1
-1	-1	-1



In the image above, it is explained that there are 2 classes consisting of -1 shown in blue and 1 shown in orange. At each of these points, it is used to find the separator between positive data and negative data.

Supervised Learning

Supervised learning is a machine learning approach in which models are trained using pre-labeled datasets. In the case of classification, a dataset consists of input examples (features) and related outputs (class labels). The goal of supervised

learning is to learn a function or model that can map the input to the correct output, so that the model can classify new examples appropriately.

Basic Principles of SVM

SVM works by searching for hyperplanes that can separate data from different classes with maximum margins. A hyperplane is a one-dimensional lower space that is used to separate data. In two-dimensional space, a hyperplane is a line, whereas in three-dimensional space, a hyperplane is a plane.

1. Hyperplane and Margin: A hyperplane is a line (or surface) that separates two classes. SVM looks for a hyperplane that maximizes margin, which is the closest distance between data from the two classes to the hyperplane. With a larger margin, the generalization of the model to the new data will be better.
2. Support Vectors: Support vectors are the data of the points that are closest to the hyperplane and are influential in determining the position of the hyperplane. Only support vectors are used in the SVM model formation process.

Function and Optimization

The purpose function of the SVM is to find a hyperplane that maximizes margins. This function can be expressed as an optimization problem with certain limitations. The SVM optimization equation for linear classification can be written as:

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2$$

With limitations:

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, \quad \forall i$$

Where:

- **w** is the weight vector.
- **b** is bias.
- x_i is the feature vector of the ith data.
- y_i is the class label of the ith data (1 or -1).

Kernel Trick

For cases where data cannot be separated linearly, SVM uses kernel tricks to map data to higher dimensional spaces where data can be separated linearly. Commonly used kernels include:

- Linear Kernel : **K (,.** $x_i\ x_j) = x_i x_j$ is a feature vector
- Polynomial Kernel : **K (,.** $x_i\ x_j) = (x_i x_j + c)^d$

- Radial Basis Function (RBF) Kernel : **K (**$x_i\ x_j) = \ exp\ (-\gamma\ \|\ x_i - x_j\ \|^2)$

RESEARCH METHOD

This study aims to implement and evaluate the performance of the Support Vector Machine (SVM) in the case of classification. The main focus is on the effectiveness of the model in separating the classes in the dataset. Here are the steps to prepare it:

Data Collection and Determination of Research Targets

The data needed is in the form of visual inspection, load reading and profilling of transformer and PHB TR subjects at UL Ciracas in March 2024. The data used is data that has been labeled Good, Enough, Lacking and Poor.

Preprocessing

In machine learning modeling, it is very important because raw data obtained from various sources is often not in a form that is ready for direct use by machine learning models.

Process Support Vector Machine (SVM)

One of the machine learning algorithms used for classification and regression tasks. In python language it can be implemented using the scikit-learn library. In this process, using python is divided into Data Training and Data Testing.

Evaluation

Using the Confusion Matrix and its visualizations, we can easily see how our SVM model is performing, including where the model makes errors in predictions. This helps in understanding the strengths and weaknesses of the classroom model.

By following this research method, the application of the Support Vector Machine in the case of classification will be carried out systematically and the results can be well evaluated.

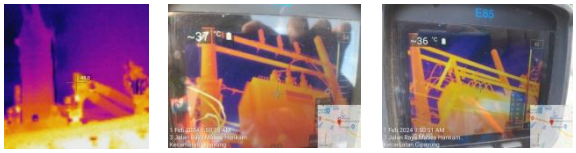
RESULT AND DISCUSSION

In this discussion, experiments were carried out for each stage along with examples of displays in Python.

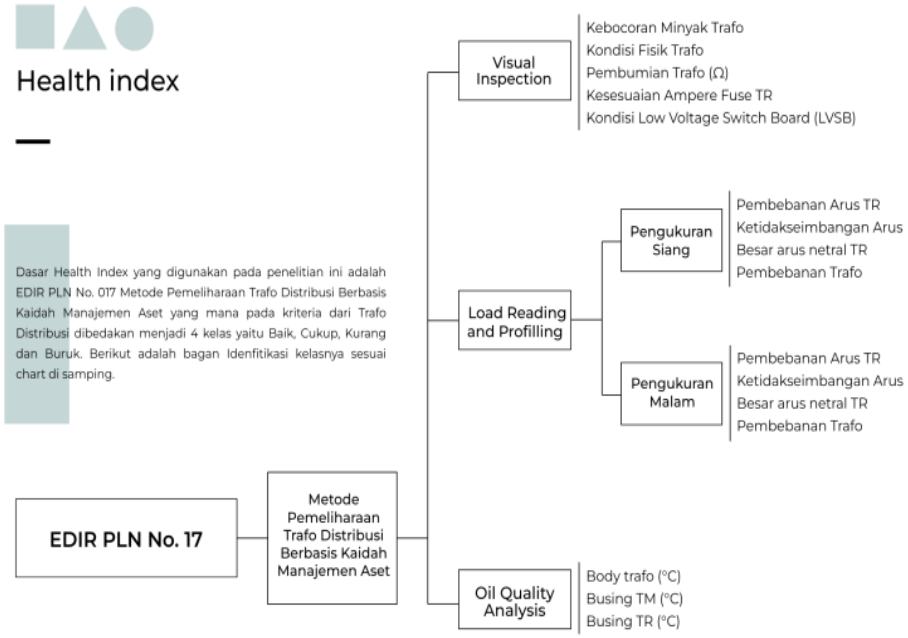
Data Collection and Determination of Research Targets

The data needed is in the form of visual inspection, load reading and profilling of transformer and PHB TR subjects at UL Ciracas in March 2024. The data used is data that has been labeled Good, Enough, Lacking and Poor. Then the data is

processed with an excell table that has been recapped. Here's an example of the measurement data. Here is an example of a picture taken:



In EDIR PLN No. 017 concerning Distribution Transformer Maintenance Methods Based on Asset Management Principles as the Basis of the Health Index requires various criteria from 1 class, the following is an overview that can be taken from the rule:



Preprocessing

1. *Editing input data*
   - Delete unnecessary columns
   - Delete null data
   - Changing the value of 'NOT AFFORDABLE' to 0
   - Turning categorical data into numerical representations
   - Preparing X&y data for use in modeling

2. *Normalization*

Scaling each feature to a specific range(0, 1), known as normalization. This is useful in some cases where the algorithm used will perform better when the

features are in a similar or standardized range. An example of normalization using the following library from Phython:

```
[18]: from sklearn.preprocessing import MinMaxScaler

# Normalisasi data
scaler = MinMaxScaler()
X_norm = scaler.fit_transform(X) # Normalisasi atribut agar memiliki skala yg sebanding

data_norm = pd.DataFrame(X_norm) # Mengubah hasil normalisasi dalam bentuk array kedalam dataframe
data_norm.columns = X.columns
data_norm
```

3. Balancing Data

Scaling each feature to a specific range (0, 1), known as normalization. This is useful in some cases where the algorithm used will perform better when its features are in a similar or standardized range as in the example below.

```
[20]: from imblearn.over_sampling import ADASYN

# Inisialisasi ADASYN
smote = ADASYN(n_neighbors = 2, random_state=42, sampling_strategy = "minority")

# Resample dataset menggunakan ADASYN
X_resampled, y_resampled = smote.fit_resample(X_norm, y)
```

```
[21]: y_resampled.value_counts().sort_index()
```

```
[21]: Kriteria
0    49
1    90
2    89
3    90
Name: count, dtype: int64
```

Process Support Vector Machine (SVM)

Support Vector Machine (SVM) in Python can be implemented using the scikit-learn library. Here is a brief explanation of how to use SVM with Python:

1. Separating Data for Training and Testing:

```
[22]: from sklearn.model_selection import train_test_split
# Membagi data manjeadi data pelatihan(train) dan data pengujian(test)
# 90% untuk data train, 10% untuk data test
X_train, X_test, y_train, y_test = train_test_split(X_resampled, y_resampled, test_size=0.1, random_state=1)
print('X_train data shape : ', X_train.shape)
print('X_test data shape  : ', X_test.shape)
```

```
X_train data shape :  (286, 16)
X_test data shape  :  (32, 16)
```

- train_test_split: This function divides the dataset into two subsets, the training set and the test set, based on the proportions determined by

the test_size parameters. In this case, 90% of the data is used for training and 10% for testing (test_size= 0.1).

- X_resampled and y_resampled: This is feature data and labels that have been resampled to address class imbalances, which are then divided into training and test sets.
- Random_state: Seed for randomization so that the data sharing can be reproduced on the next experiment.

2. SVM Model Implementation

```
[23]: from sklearn.svm import SVC
      from sklearn.model_selection import GridSearchCV

      param_grid = {'C': [1, 10, 100, 1000, 10000],
                    'gamma': [1, 0.1, 0.001, 0.0001],
                    'kernel': ['linear','rbf']}
      Svm = GridSearchCV(SVC(), param_grid=param_grid)
      Svm.fit(X_train, y_train)

      # Menampilkan kombinasi parameter terbaik yang diperoleh
      print('Best parameter : ', Svm.best_params_)

      pred_svm = Svm.predict(X_test)

      Best parameter :  {'C': 1000, 'gamma': 1, 'kernel': 'linear'}
```

In the above code, the Support Vector Machine (SVM) of the sklearn library is used to build a classification model with grid search (GridSearchCV) to optimize the SVM hyperparameter parameters. Here is a detailed explanation of the parameters in the code:

- 'C': This parameter is a regulation of parameters that determines the trade-off between achieving large margins and ensuring data points are properly classified. The values tested are [1, 10, 100, 1000, 10000].
- 'gamma': This parameter is only used for the 'rbf' kernel. Gamma defines how far the influence of a single training point is. A small value means a larger 'radius' of influence, while a large value means a smaller influence. The values tested were [1, 0.1, 0.001, 0.0001].
- 'kernel': This parameter specifies the type of kernel that the SVM algorithm will use. The kernel can convert the non-linear input space into a higher feature space to make it easier to separate the data points. The values tested are ['linear', 'rbf'].

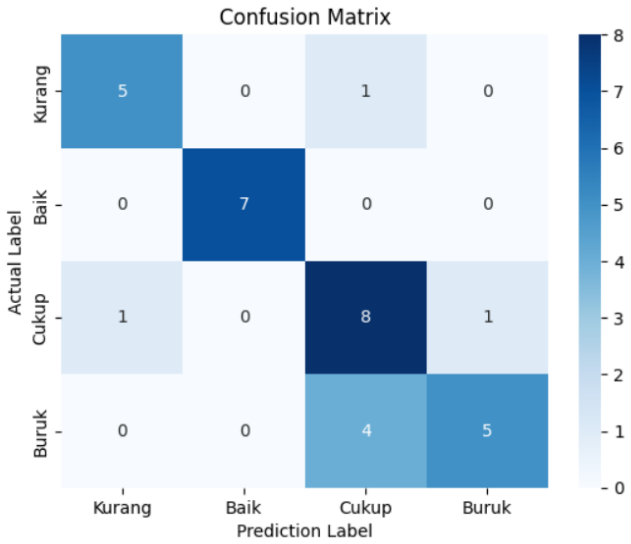
Evaluation

```
[24]: import seaborn as sns
      import matplotlib.pyplot as plt
      from sklearn.metrics import accuracy_score, confusion_matrix, precision_score, recall_score, f1_score

      Confusion_matrix = confusion_matrix(y_test, pred_svm)
      class_label      = ['Kurang', 'Baik', 'Cukup', 'Buruk']
      df_confusion     = pd.DataFrame(Confusion_matrix, index = class_label, columns = class_label)

      sns.heatmap(df_confusion, annot=True, fmt = "d", cmap=plt.cm.Blues)
      plt.title('Confusion Matrix')
      plt.xlabel('Prediction Label')
      plt.ylabel('Actual Label')
      plt.show()
```

From the coding it produces the following confusion matrix image:



From the tested testing data, there are:

1. The prediction results for the Less class were 6 samples that matched 1 sample and did not match 1 sample (Enough class).
2. The prediction results for the Good class were 7 samples, all of which were in accordance with the actual label.
3. The prediction results for the Sufficient class were 13 samples that matched 8 samples and did not match 5 samples (1 in the Poor class, 4 in the Poor class).
4. The prediction results for the Bad class were 6 samples that matched 5 samples and did not match 1 sample (Sufficient class).

```
[25]: accuracy_svm  = accuracy_score(y_test, pred_svm)
      precision_svm = precision_score(y_test, pred_svm, average='weighted')
      recall_svm    = recall_score(y_test, pred_svm, average='weighted')
      f1_score_svm  = f1_score(y_test, pred_svm, average='weighted')

      print('Accuracy  : ',round(accuracy_svm*100, 2), '%')
      print('Precision : ',round(precision_svm*100, 2), '%')
      print('Recall    : ',round(recall_svm*100, 2), '%')
      print('F1-score  : ',round(f1_score_svm*100, 2), '%')

Accuracy  :  78.12 %
Precision :  80.17 %
Recall    :  78.12 %
F1-score  :  77.99 %
```

From the coding above, the functions of each library can be explained as follows:

1. The function accuracy_score calculates the accuracy of the model, which is the correct proportion of predictions from the total predictions. Accuracy is the simplest metric and shows how well the model is at classifying samples correctly.
2. The function precision_score calculate the precision model. Precision is the ratio between the number of correct positive predictions and the total positive predictions made by the model. The average='weighted' parameter is used to calculate a weighted mean of precision with the proportion of each class in the data, which is useful when working with unbalanced datasets.
3. The function recall_score calculates model recalls. Recall is the ratio between the number of correct positive predictions and the total number of actual positive samples. The average='weighted' parameter is used to calculate the weighted average of recalls as a proportion of each class in the data.
4. The function f1_score calculate the F1-score model. The F1-score is the harmonic average of precision and recall, providing a balance between the two. F1-score is useful when we need a balance between precision and recall, especially when the dataset is unbalanced.

CONCLUSION

From the research that has been carried out, several results have been obtained with variations of 60%, 70%, 80% and 90% training data as follows:

No.	Uraian Evaluasi	Data Latih			
		60%	70%	80%	90%
1	accuracy	76.56%	73.96%	75.0%	78.12%
2	precision	76.57%	74.99%	74.96%	80.17%
3	recall	76.56%	73.96%	75.0%	78.12%
4	F1-score	75.7%	73.43%	74.62%	77.99%

This study shows that the Support Vector Machine (SVM) has high effectiveness in classification tasks, especially when the data has a clear separation margin. Choosing the right kernel has a significant impact on SVM performance, with linear kernels suitable for data that can be separated linearly, while non-linear kernels such as RBF are more effective for data with more complex patterns. In addition, data pre-processing such as normalization or standardization proved to be important to ensure the stability of results. Hyperparameter tuning, such as C and γ parameters, also plays a big role in improving model accuracy, where techniques such as Grid Search and cross-validation help find optimal combinations.

Compared to other algorithms such as Decision Tree or Neural Networks, SVMs often show competitive performance, especially on high-dimensional data. Its excellence in maximizing the separation margin makes it a solid choice in most classification cases. In addition, the aspects of transparency and reproducibility in SVM research are essential, where good documentation can help further research. With these findings, this study is expected to provide deeper insights into the application of SVM and open up opportunities for further research that is more applicable.

REFERENCES

A. T. Islam, M. A. Rashid, M. N. Hasan, and M. R. Hasan, "A Comparative Study of Support Vector Machine and Naive Bayes Classifier for Spam Email Detection," 2020 International Conference on Computing and Information Technology (ICCIT), Dhaka, Bangladesh, 2020, pp. 1-5, doi: 10.1109/ICCIT-144147.2020.9325354.

P. Kumar, R. Singh, and M. S. Deshpande, "Support Vector Machine Based Fault Diagnosis in Rotating Machinery Using Sound Signal," 2021 IEEE International Conference on Automation Science and Engineering (CASE), Lyon, France, 2021, pp. 1-6, change: 10.1109/CASE49439.2021.9676748.

J. Zhang, H. Li, and Y. Wang, "Support Vector Machine Classification Algorithm and Its Application in Hyperspectral Image Classification," 2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS, Brussels, Belgium, 2021, pp. 1-4, change: 10.1109/IGARSS47720.2021.9553385.

X. Li, Y. Zhang, and T. Chen, "A Study of Support Vector Machine for Intrusion Detection System," 2021 13th International Conference on Computer and Automation Engineering (ICCAE), Melbourne, Australia, 2021, pp. 1-6, change: 10.1109/ICCAE51655.2021.9376012.

R. S. Mufid, E. M. Margiana, and T. S. Anggoro, "Application of Support Vector Machine (SVM) for Heart Disease Prediction with Different Kernel Functions," 2021 International Conference on Artificial Intelligence and Data Sciences (AiDAS), Malang, Indonesia, 2021, pp. 1-5, doi: 10.1109/AiDAS53897.2021.9570564.

N. Patel and H. Prajapati, "Comparative Analysis of Support Vector Machine and Random Forest Classification Techniques in the Prediction of Diabetes," 2021 6th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2021, pp. 1-5, change: 10.1109/ICCES51350.2021.9489196.