JUSIKOM PRIMA (Journal of Information Systems and Computer Science Prima)
Vol. 6 No. 2, February 2023 E-ISSN: 2580 - 2879

IMPLEMENTATION OF SUPPORT VECTOR MACHINE ALGORITHM WITH HYPER-TUNING RANDOMIZED SEARCH IN STROKE PREDICTION

Yennimar[1], Alvin Rasid[2], Sun Kenedy[3]
Universitas Prima Indonesia
Jl. Sampul No.4, Sei Putih Bar., Kec. Medan Petisah, Medan City, North Sumatra 20118
E-mail:yennimar@unprimdn.ac.id

ABSTRACT- Stroke is a severe health problem and can significantly impact a person's quality of life. Therefore, it is crucial to predict stroke early so that preventive measures can be taken before it is too late. This study demonstrates the importance of hyper tuning and hyperparameters in a stroke prediction model. Literature studies show that many studies on stroke prediction need to explain this, even though this is very important for developing the performance of stroke prediction models. In this study, we use the Support Vector Machine (SVM) algorithm to predict stroke and evaluate the algorithm's performance without hyper tuning and with hyper tuning Randomized Search CV. We also divide the data into training and test data by 75% and 25%. The results of this study indicate that hyper-tuning can improve the accuracy of the stroke prediction algorithm. The algorithm's accuracy is 77% without hyper-tuning, whereas, with hyper-tuning, the accuracy increases to 96%. Hypertuning with the Randomized Search CV method can improve the performance of the stroke prediction algorithm and is very important to do in developing predictive models.

Keywords: Stroke Prediction; Binary Classification; Support Vector Machines; Randomized Search CV

1. INTRODUCTION

Stroke is a disease caused by impaired blood flow to the brain and is the second most common cause of death and disability worldwide. Therefore, early diagnosis is crucial to prevent further damage[1]. Machine Learning technologies, such as the Random Forest algorithm[2], Support Vector Machine (SVM)[7], [9], Artificial Neural Networks (ANN)[4], and Decision Trees[4], can be used to predict stroke. For example, a study by Wang et al. [4]found that SVM performs better than other algorithms. However, several related studies do not describe Hyper Parameters and Hypertuning, which can help improve accuracy performance[4].

Three Hypertuning techniques are widely used for similar prediction problems, namely Randomized Search[5]–[7], GridSearch [7]–[9], and the Tree Base Pipeline Optimization Tool[8], [10], [11]. Previous research found that the Randomized Search technique produces the best performance compared to other courses on medical datasets such as Cleveland heart disease and Z-Alizadeh Sani.

On the other hand, the Stroke Dataset from the Medical Clinic of Bangladesh (MCB) has been extensively studied with the SVM algorithm and the SMOTE method to address the Data Imbalance problem. Still, the benefits of Randomized Search have yet to be explored.[2], [12]–[15]. Therefore, this study will examine the application of the SVM algorithm with and without hyper tuning Randomized Search CV.

2. Method

2.1. Stroke Prediction

Stroke Prediction aims to classify stroke in patients. The design flow of the stroke prediction model using the Machine Learning Support Vector Machine (SVM) algorithm includes:

1. Enter training data input in the form of patient data with stroke patient data and those without
2. The SVM algorithm will carry out the model training process and produce a trained SVM model to predict stroke
3. Enter patient testing data input into the trained SVM model
4. The SVM model produces predictive output whether the patient has a stroke or not

2.2. Support Vector Machine (SVM)

Support Vector Machine (SVM) is a Machine Learning algorithm that works by mapping data into high-dimensional feature spaces to categorize data points. Even though data cannot be separated linearly, separators or separators between categories can be found, as shown in the following figure.

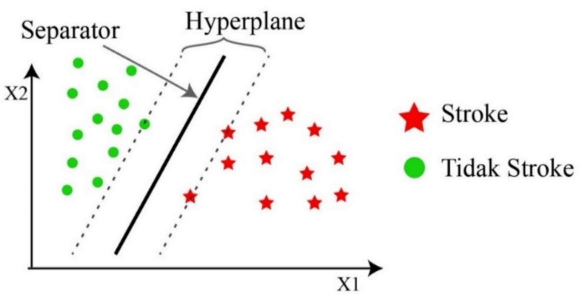


Figure 1. Illustration of a Support Vector Machine

Then the data is transformed so that the separator can become a hyperplane. Then, the characteristics of the new data can be used to predict which groups should have new records [4].

2.3. Randomized Search

Randomized Search (RS) is a technique in which a random combination of Important Parameters and Unimportant Parameters is used to find the best solution for the model being built [10]. As shown in the image above, this method is similar to Grid Search (GS), except that the way the GS method searches resembles a grid pattern and is not random.

2.4. Datasets

The material to be used in this study is the Stroke dataset from the Medical Clinic of Bangladesh (MCB).[1]which has been investigated in various previous studies [4]–[8] with the following specifications:

Table 1. Dataset Specifications

Specification	Information
Number of data (N)	5110 data
Independent variable (x)	Represents patient's personal and medical data, totaling 11 variables
dependent variable (y)	Stroke or No Stroke
Number of negative data (ny=0)	4861
Number of positive data (ny=1)	249

2.5. Research procedure

The procedure for this research follows the general Machine Learning procedure as shown in the following flowchart:

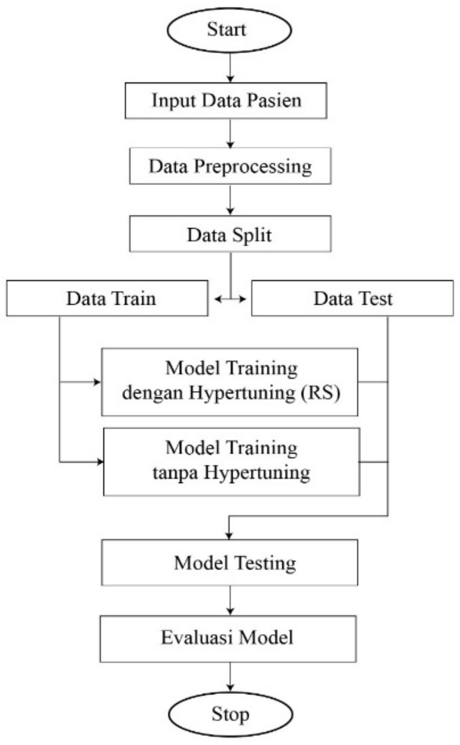


Figure 2. Research Flow

The stages, as shown in the image above, include:

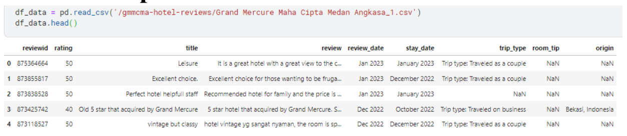
1. Patient Data Input. This process enters data in Comma Separated Values (CSV) format, including personal data, medical records, etc.
2. Data Preprocessing. This process processes and transforms data to be processed correctly by SVM algorithms, including Label Encoding and the Synthetic Minority Over-Sampling Technique (SMOTE) Method to handle data imbalances[16].
3. Split Data. This process divides the data into Data Train and Data Test.
4. Training Model Without Hypertuning. This process trains the SVM model to predict stroke with Data Train.
5. Model Training With Hypertuning (RS). The difference with Hypertuning is that it uses the Randomized Search (RS) method to improve model performance.
6. Testing Models. This process tests the SVM model trained to predict stroke with the Data Test.
7. Evaluation. The last process is to evaluate the performance of the model by calculating the accuracy of the Machine Learning model by knowing the four possible results that occur, namely True Negative (TN), True Positive (TP), False Negative (FN), and False Positive (FP). The

four results are also called the Confusion Matrix. Then the calculation of the accuracy value can be done like the following formula:

$$Accuracy = \frac{(TP+TN)}{(TP+FN+TN+FP)} \quad (1)$$

3. Results

3.1. Data inpatient



```
df_data = pd.read_csv('/gmmcma-hotel-reviews/Grand Mercure Maha Cipta Medan Angkasa_1.csv')
df_data.head()
```

```
    reviewid  rating                                title                                           review  review_date       stay_date                        trip_type  room_tip             origin
0  875364664      50                              Leisure    It is a great hotel with a great view to the c...     Jan 2023    January 2023  Trip type: Traveled as a couple       NaN                NaN
1  873855817      50                     Excellent choice.  Excellent choice for those wanting to be fruga...     Jan 2023   December 2022  Trip type: Traveled as a couple       NaN                NaN
2  873838528      50           Perfect hotel helpfull staff  Recommended hotel for family and the price is ...     Jan 2023    January 2023                              NaN       NaN                NaN
3  873425742      40  Old 5 star that acquired by Grand Mercure   5 star hotel that acquired by Grand Mercure. S...     Dec 2022    October 2022  Trip type: Traveled on business       NaN  Bekasi, Indonesia
4  873118527      50                     vintage but classy  hotel vintage yg sangat nyaman, the room is sp...     Dec 2022   December 2022  Trip type: Traveled as a couple       NaN                NaN
```

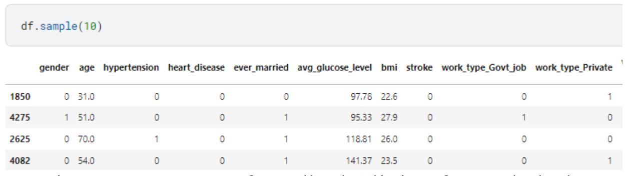
Figure 3. Dataset of Medical Clinic of Bangladesh (MCB) in CSV form

Patient data is initially received in CSV (Comma Separated Values) format, which is the format often used to store datasets. However, this format is not suitable for data analysis. Therefore, we migrated the patient data from CSV to Jupyter Notebook Dataframe format. Dataframe is a table data representation in matrix form, which makes it easier for researchers to visualize and analyze data. Using data frames, researchers can perform various operations such as filtering, data preprocessing, etc. In addition, the patient data input process helps researchers prepare data for further analysis.

3.2. Data Preprocessing

Data preprocessing is a crucial stage in research, where the data collected must be processed and prepared for analysis. The researcher performs the label encoding process on the patient data in this case. Label encoding converts categorical data (such as gender, blood type, etc.) into numbers to facilitate data analysis. For example, the gender "Female" can be coded as number 1, and the gender "Male" can be coded as number 0. This allows the Machine Learning analysis algorithm used in this research to understand better and process the data.

Thus, the label encoding process assists researchers in preparing patient data for further analysis and produces a ready-to-process Dataframe, as shown in the following figure.

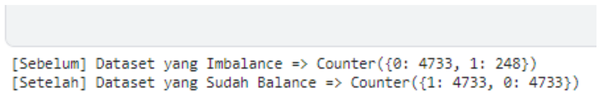


```
df.sample(10)
```

```
      gender   age  hypertension  heart_disease  ever_married  avg_glucose_level   bmi  stroke  work_type_Govt_job  work_type_Private
1850       0  31.0             0              0             0              97.78  22.6       0                   0                  1
4275       1  51.0             0              0             1              95.33  27.9       0                   1                  0
2625       0  70.0             1              0             1             118.81  26.0       0                   0                  0
4082       0  54.0             0              0             1             141.37  23.5       0                   0                  1
```

Figure 4. Dataset of Medical Clinic of Bangladesh (MCB) in CSV form

In addition, other problems need to be addressed, namely Imbalanced data. This happened because the data on stroke patients amounted to only 248, while there were fewer than 4733. In this study, we used the SMOTE method to overcome this problem resulting in 4733 data on stroke patients and those who did not have strokes, so the total data became 9466, as shown in the following figure.



```
[Sebelum] Dataset yang Imbalance => Counter({0: 4733, 1: 248})
[Setelah] Dataset yang Sudah Balance => Counter({1: 4733, 0: 4733})
```

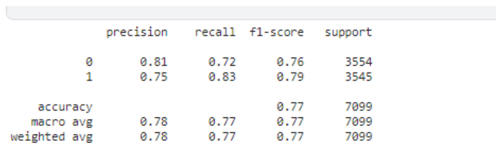
Figure 5. Correcting Imbalanced Data

3.3. Split Data

In this study, patient data were processed through data input and preprocessing processes, divided into two major parts. In the first part, 75% is used as training data or is called Data Train, while in the second part, 25% is used as test data or is called Data Test. This aims to ensure the algorithm can work well on new data that has never been seen before.

3.4. Model TrainingNo Hypertuning

Hypertuning is the process of optimizing the parameters in a model so that the model performance improves. However, in this study, the model for determining the results of a diagnosis using the SVM (Support Vector Machine) algorithm will be distinguished between those using the hyper tuning and those not using hyper tuning. Data Train, previously divided into 75% parts, is used to train the SVM model. This model is implemented by entering training data and conducting training using the algorithm. After training, the model will be ready to predict stroke with Test data.



```
              precision    recall  f1-score   support

           0       0.81      0.72      0.76      3554
           1       0.75      0.83      0.79      3545

    accuracy                           0.77      7099
   macro avg       0.78      0.77      0.77      7099
weighted avg       0.78      0.77      0.77      7099
```

Figure 6. Evaluation of Training Performance Without Hypertuning

3.5. Model TrainingWith Randomized Search Hypertuning

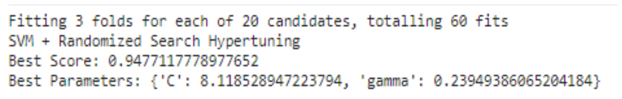
As previously explained, parameters must be presented in research reports to ensure research results can be reused and shared findings with other researchers. In addition, the code used is also attached on the last page of this research report. Next, we will describe the first parameter, namely the Randomized Search Hypertuning parameter used as follows:

Table 2. Randomized Search Parameters

Parameter Type Randomized Search	Parameter Value
param_distributions	'C' : [2, 4, 6, 8] 'gamma' : [0.1, 0.3, 0.5, 0.7, 0.9]
n_iter	20
n_jobs	4
cv	3

random_state	42
scoring	from sklearn. metrics \import make_scorer, roc_auc_score make_score(roc_auc_score)

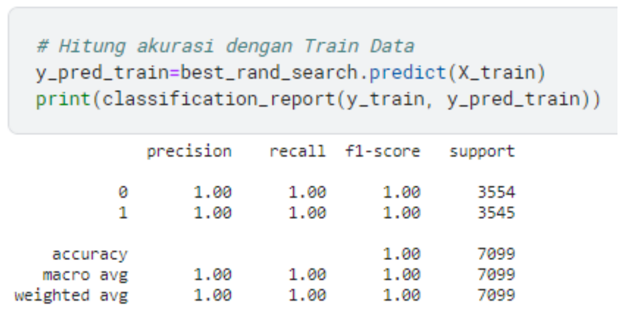
After doing Hypertuning, the best hyperparameter results that the SVM model can use to predict stroke in this study are as follows:



```
Fitting 3 folds for each of 20 candidates, totalling 60 fits
SVM + Randomized Search Hypertuning
Best Score: 0.9477117778977652
Best Parameters: {'C': 8.118528947223794, 'gamma': 0.23949386065204184}
```

Figure 7. Best Hyperparameter Results

Then, evaluating the performance of the training process with hyper-tuning can be seen in the following figure:



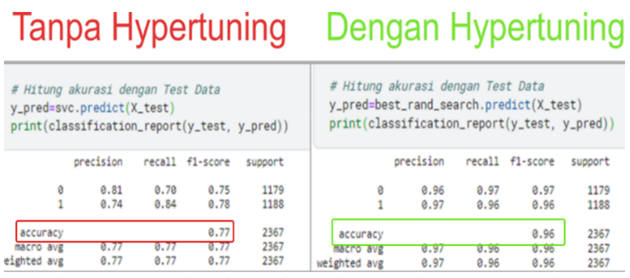
```
# Hitung akurasi dengan Train Data
y_pred_train=best_rand_search.predict(X_train)
print(classification_report(y_train, y_pred_train))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      3554
           1       1.00      1.00      1.00      3545

    accuracy                           1.00      7099
   macro avg       1.00      1.00      1.00      7099
weighted avg       1.00      1.00      1.00      7099
```

Figure 8. Evaluation of Training Performance with Hypertuning

It can be seen that the results of training with hyper-tuning reach a perfect score, but this does not mean that the model is feasible for testing and needs to be proven again with the following process, namely testing.

3.6. Testing Models



Tanpa Hypertuning

```
# Hitung akurasi dengan Test Data
y_pred=svc.predict(X_test)
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.81      0.70      0.75      1179
           1       0.74      0.84      0.78      1188

    accuracy                           0.77      2367
   macro avg       0.77      0.77      0.77      2367
eighted avg       0.77      0.77      0.77      2367
```

Dengan Hypertuning

```
# Hitung akurasi dengan Test Data
y_pred=best_rand_search.predict(X_test)
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.96      0.97      0.97      1179
           1       0.97      0.96      0.96      1188

    accuracy                           0.96      2367
   macro avg       0.97      0.96      0.96      2367
weighted avg       0.97      0.96      0.96      2367
```

Figure 9. Evaluation of Testing Performance Using Hypertuning

This study tested the model with and without hyper-tuning, as shown in the following figure. The model's results without hyper tuning show good results with an accuracy of 77%. Precision and recall for class 0 (no stroke) and class 1 (stroke) were 0.81 and 0.74 and 0.70 and 0.84, respectively. However, after hyper tuning the model, the results obtained improved with an accuracy of 96%. Precision and recall for class 0 and class 1 were 0.96 and 0.97 and 0.97 and 0.96, respectively. This shows that by hyper tuning the model, the results obtained are better and more accurate.

4. Conclusion

Some of the conclusions from this study are as follows:

1. Based on the latest literature study, this study found that stroke prediction algorithms can be helpful for technological developments in the medical world.
2. However, many studies need to share information on hyperparameters and hyper-tuning, which are used to improve the accuracy of stroke prediction. Therefore, this study also provides examples of information related to hyperparameters and hyper tuning used.
3. This study shows that the Support Vector Machine (SVM) algorithm can be used to predict stroke risk in patients. The results of the training model without hyper-tuning show an accuracy of 77%, whereas, with Randomized Search Hypertuning, it offers an increase in accuracy to 96%. This indicates that hyper-tuning plays an essential role in increasing model accuracy.

Based on the results of this study, we suggest the following:

1. First, conduct further research by adding new variables and comparing other algorithms.
2. Conduct clinical studies to test the validity of developed models on natural populations or other datasets.
3. Because this study did not use Cross Validation, further research can use the Cross Validation method to validate the model results further.

BIBLIOGRAPHY

[1] GA Roth, "Global Burden of Disease Collaborative Network. Global Burden of Disease Study 2017 (GBD 2017) Results. Seattle, United States: Institute for Health Metrics and Evaluation (IHME), 2018, "*The Lancets*, vol. 392, pp. 1736–1788, 2018.

[2] Z. Li, "Study of ICU Mortality Prediction and Analysis based on Random Forest," in *Proceedings of the 7th International Conference on Cyber Security and Information Engineering*, 2022, pp. 691–695.

[3] PPremisha, S. Prasanth, M. Kanagarathnam, and K. Banujan, "An Ensemble Machine Learning Approach for Stroke Prediction," in 2022 International Research Conference on Smart Computing and Systems Engineering (SCSE), 2022, vol. 5, pp. 165–170.

[4] W. Wang*et al.*, "A systematic review of machine learning models for predicting outcomes of stroke with structured data," PLoS One, vol. 15, no. 6, p. e0234722, 2020.

[5] QBadriyah, N. Sakinah, I. Syarif, and DR Syarif, "Machine learning algorithm for stroke disease classification," in 2020 International Conference on

Electrical, Communication, and Computer Engineering (ICECCE), 2020, pp. 1–5.

[6] NK Al-Shammari et al., “Cardiac stroke prediction framework using hybrid optimization algorithm under DNN,” Engineering, Technology & Applied Science Research, vol. 11, no. 4, pp. 7436–7441, 2021.

[7] S. Bhattacharya *et al.*, “Antlion re-sampling based deep neural network model for classification of imbalanced multimodal stroke dataset,” Multimed Tools Appl, pp. 1–25, 2020.

[8] R.Valarmathi and T. Sheela, "Heart disease prediction using hyperparameter optimization (HPO) tuning," Biomed Signal Process Control, vol. 70, p. 103033, 2021.

[9] J. -M. Choi*et al.*, “Prediction of Hemorrhagic Transformation after Ischemic Stroke Using Machine Learning,” J Pers Med, vol. 11, no. 9, p. 863, 2021.

[10] V. Nagarajan and V.Vijayaraghavan, “End-to-end optimized arrhythmia detection pipeline using machine learning for ultra-edge devices,” arXiv preprint arXiv:2111.11789, 2021.

[11] BCFati, A. Muneer, NA Akbar, and SM Taib, “A continuous cuffless blood pressure estimation using a tree-based pipeline optimization tool,” Symmetry (Basel), vol. 13, no. 4, p. 686, 2021.

[12] R.Mitraa and T. Rajendranb, “Efficient Prediction of Stroke Patients Using Random Forest Algorithm in Comparison to Support Vector Machine,” 2022.

[13] F. Akbar, HWSaputra, AK Maulaya, MF Hidayat, and R. Rahmaddeni, "Implementation of the C4 Decision Tree Algorithm. 5 and Support Vector Regression for Prediction of Stroke: Implementation of Decision Tree Algorithm C4. 5 and Support Vector Regression for Stroke Disease Prediction,” MALCOM: Indonesian Journal of Machine Learning and Computer Science, vol. 2, no. 2, pp. 61–67, 2022.

[14] A. Gaffney, “An Ensemble Learning Algorithm for ICU Patient Mortality Prediction,” *Doctoral dissertation, Dublin, National College of Ireland*, 2021.

[15] S. Cohen, N. Dagan, N. Cohen-Inger, D.Ofer, and L. Rokach, “ICU Survival Prediction Incorporating Test-Time Augmentation to Improve the Accuracy of Ensemble-Based Models,” IEEE Access, vol. 9, pp. 91584–91592, 2021.

[16] M. Ghosh, “An Enhanced Stroke Prediction Scheme Using SMOTE and Machine Learning Techniques,” in *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, 2021, pp. 1–6.