

PREDICTION OF ELECTRICITY USE USING A WEBSITE-BASED SUPPORT VECTOR MACHINE ALGORITHM

Reyhan Adiptya¹, Muhammad Ary Murti², Casi Setia Ningsih³

^{1,2,3} Telkom University, Bandung

reyhanadiptya@student.telkomuniversity.ac.id¹, arymurti@telkomuniversty.ac.id²,

setiacasie@telkomuniversty.ac.id³

Abstract— This study aims to create an electrical load prediction system using the Support Vector Machine algorithm to be able to predict future electrical loads. This study also finds out what parameters can reduce the error rate of predictions using Particle Swarm Optimization. Then everything is packaged into a website using the flask framework. The results of testing the parameters of the Support Vector Machine algorithm on the electricity usage prediction system, the lowest error values obtained are MAE, MSE, RMSE on the parameters of the PSO optimization results, the SVR parameter value is $C = 1$; $\text{Gamma}=8.3$; $\text{Epsilon}=0.001$; produces an error value, $\text{MAE}=0.00829921$; $\text{MSE}=0.00602241$; $\text{RMSE}=0.0776042$.

Keywords—Support Vector Machine, Particle Swarm Optimization, Prediction, Penggunaan Energi Listrik

I. INTRODUCTION

Prediction of electricity use is one way that can be done to manage and distribute energy economically and efficiently. The results of this prediction of electricity use can be used to control and prevent inefficient electricity use in the following month [1]. Support Vector Machines (SVM) is a technique for making predictions, both in the case of classification and regression. The use of the SVM model processes data into training data and test data. The training data is used in forming the SVM model. For the application itself, a website is also made in which there is already a program that accepts input data and produces output predictions of electricity usage.

II. RELATED WORK

Previous electricity usage prediction studies generally use the Neural Network method. Neural Network is an information processing system that has characteristics similar to biological neural networks, however, the shortcomings of Neural Networks are often overfitting because they are overtrained [2]. Then there is also research that uses the Naive Bayes method. However, the Naive Bayes method has several drawbacks, namely: The assumption that each independent variable reduces accuracy; because there is usually a correlation between one variable and another; Its accuracy cannot be measured using a single probability. Need other evidence to prove it [3].

III. PROPOSED METHOD

In this section we will focus on the methods that have been developed for the system. The system created can predict electrical energy in the following month and is packaged in the form of a website.

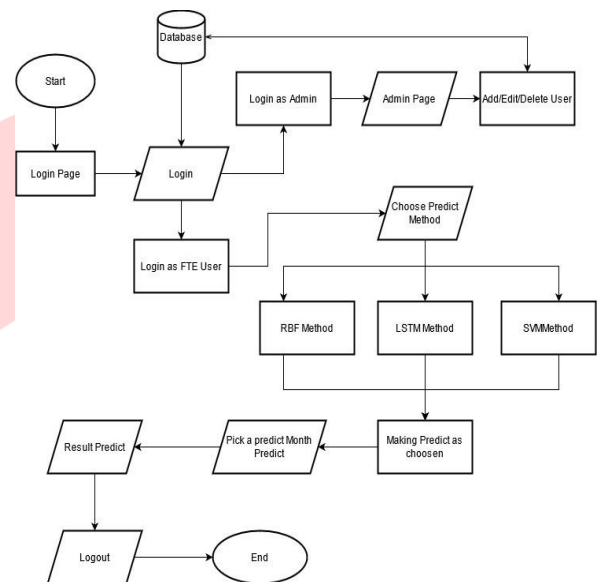


Figure 1 Flowchart Sistem

Figure 1 explains that before making predictions the user is expected to log in first. After logging in based on the registered username, each user can make predictions based on the user role registered by the admin. The admin role can only do user management. As for FTE users, predictions are based on a dataset that has been prepared by the admin. On the prediction page, each user has the same feature, namely the selection of prediction dates. On the prediction page there are also features of the average Kwh usage per month, per week, per day, and a graph of the data. Users can make predictions based on the selection of the desired date.

1. Data

The data taken comes from the electric current reader device in the P Telkom University building. It should be noted that there are data with low energy use due to the presence of solar panels that also supply electricity to the P building. The data read by the device only records the use of electricity sourced from PLN.

2. Preprocessing

The dataset used is the dataset of the electric reading device which is placed in the P Telkom University building. This dataset is then retrieved from ANTARES using an API which will be sent to the Google Colab IDE with the help of the JSON programming language. Data can be retrieved from the device API which contains the recorded date, time, device number and Kwh.

Table1 Raw Data

	Kwh	DateTime
28775	401	7/8/2021 14:42
28776	401	7/8/2021 14:47
28777	401	7/8/2021 14:52
28778	401	7/8/2021 14:57
28779	401	7/8/2021 15:02

From the total existing dataset, there is data that does not fit the format, so it needs to be adjusted to the overall data. The data should be filtered and discarded data that is dated incorrectly. The format used is "Years – Month – Day Hour : Minutes : Second". Then data that has a Kwh value of 0 will be discarded so as not to interfere with the data prediction process during the training process. Then sample the electrical data so that it has hourly records. To calculate the hour sample, use the following equation:

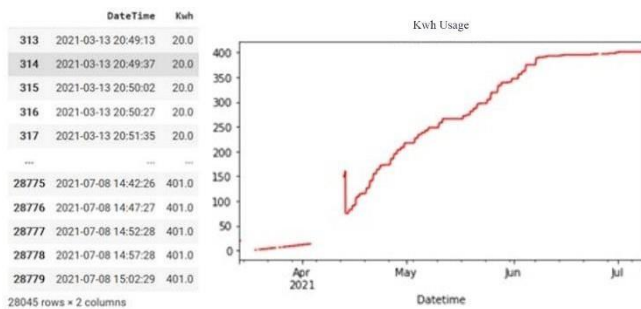
$$\text{Hour}_n = \text{MAX}(\text{Time}_{i-1}, \text{Time}_{i-2}, \dots, \text{Time}_{i-n}) \quad (1)$$

Desc:

Hour_n = Data Searching Hour

Time_{i-1}, = Data time in range each Hour

Table 2 Resample Data Each Hour



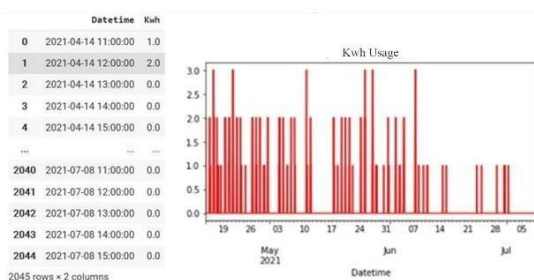
Langkah selanjutnya adalah mengisi data yang kosong dengan metode interpolasi linier:

$$y = y_1 + \frac{(x-x_1)}{(x_2-x_1)} \times (y_2 - y_1) \quad (2)$$

Because there may be an error on the device, a data anomaly is found which indicates that the data is not recorded properly, so the remaining 70% of the data is used. Then need to find the value of the interval between the current value and the next value. This is done to obtain stationary electrical data. Here is the mathematical function.

$$X = X_{t+1} + X_t \quad (3)$$

Table 3 Transform to Data Interval



The data used is data that has been converted into stationary form which aims to get good results in the prediction process. After that, the data obtained was changed to supervised form by taking samples of the first 10 data. The goal is changed into a supervised form so that the number of nc/centroids can be adjusted to the number in the model. Then the data is divided into training data and testing data.

Table 4 Supervised data

	var1(t-10)	var1(t-9)	var1(t-8)	var1(t-7)	var1(t-6)	var1(t-5)	var1(t-4)	var1(t-3)	var1(t-2)	var1(t-1)	var1(t)
10	0.0	0.0	0.0	1.0	1.0	1.0	1.0	0.0	0.0	0.0	1.0
11	0.0	0.0	1.0	1.0	1.0	1.0	0.0	0.0	0.0	1.0	0.0
12	0.0	1.0	1.0	1.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0
13	1.0	1.0	1.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
14	1.0	1.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
...
1660	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1661	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1662	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1663	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1664	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

1655 rows x 11 columns

3. Parameter Initialization

Initialization of parameters needed in the Support Vector Regression method is the value of C, gamma, and epsilon. The value of the SVR parameter will be determined using Particle Swarm Optimization to determine the value of C, gamma, and epsilon. In PSO optimization there are also required parameters, namely the number of particles, iterations, dimensions, and inertia.

a. Particle Swarm Optimization

The Particle Swarm Optimization Algorithm, abbreviated as PSO, was first introduced by Dr. Eberhart and Dr. Kennedy in 1995 at a neural network conference in Perth, Australia [4]. The particle swarm optimization algorithm is a stochastic-based optimization technique inspired by the social behavior of a flock of birds or a group of fish [5]. This optimization method based on swarm intelligence is called a behaviorally inspired algorithm as an alternative to genetic algorithms, which is often called evolution-based procedures. In the context of multivariable optimization, the swarm is assumed to have a certain or fixed size with each particle starting position at a random location in multidimensional space. Each particle is assumed to have two characteristics: position and velocity. Each particle moves in a certain space and remembers the best position ever traversed or found with respect to the food source or the value of the objective function. Each particle conveys information or good position to other particles and adjusts the position and speed of each based on the information received about the good position. Particles in PSO never die.

The particle can be viewed as a simple agent that flies through the search space and stores the best solution found. The question "How do particles fly from one position to another?" This is done simply by adding the v-vector to the x-vector to get a new x-vector, which is defined as $X_i = X_i + V_i$. Once the particle gets a new X_i it will evaluate its new position. If x-fitness is better than p-fitness, then $P_i = X_i$ and p-fitness = x-fitness. PSO starts with a randomly generated set of particles. Each particle is then evaluated for quality using the fitness function. Furthermore, the particles will fly following the optimum particles. In each generation, each particle is updated following the two "best" values. The first is the best fitness achieved by a single particle at this time. The best fitness value is denoted by p and stored in memory. While

the second "best" value is the best fitness achieved by all particles in the adjacency topology. The g index is used to designate the particle with the best fitness. If you use a neighboring topology in the form of a ring topology, then this method is referred to as the local version of PSO. However, if the neighboring topology is a star topology then this method is called the global version of PSO. The following is a mathematical model that describes the speed and position updating mechanism [6]:

$$Vi(t) = Vi(t - 1) + (c1r1(pbest - xi(t - 1)) + c2r2(gbest - Xi(t - 1)) \quad (4)$$

$$Xi(t) = Vi(t) + Xi(t - 1) \quad (5)$$

Desc:

V = Particle Velocity W = Inertia Load

C1 = Cognitive Load C2 = Social Load

Rand = Random function

Pbest = the best position possessed by a particle

Gbest = the best position globally

c1 and c2 are constants that are positive which is usually referred to as the learning rate. Then r1 and r2 are random numbers with values between 0 to 1. Equation (4) is used to calculate the new particle velocity based on the previous velocity, the distance between the current position and the local best position of the particle, and the distance between the current position. with the best position of the herd (global best). Then the particle flies to a new position based on equation (5). After the PSO algorithm is run with a certain number of iterations until it reaches the termination criteria, a solution that lies in the global best will be obtained. Keep in mind that individual best positions and group best positions need to be saved for the entire iteration. The particle content in PSO is weight. The PSO algorithm includes the following steps [6]:

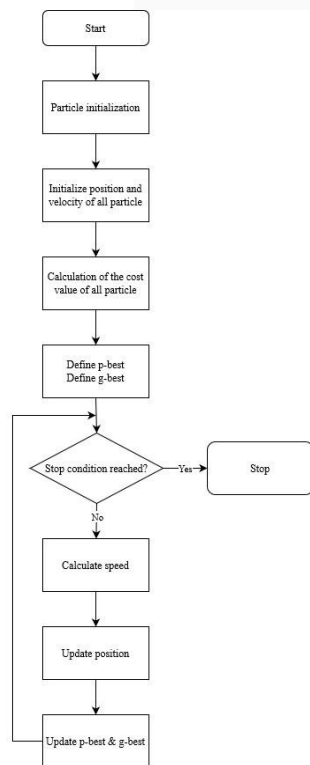


Figure 2 Flowchart PSO

PSO optimization is used to find the best SVR parameters. The parameters that PSO is looking for are C, gamma, and epsilon. Therefore the dimensions used are 3 dimensions. For the number of particles suggested from 20 to 40 [7]. For iteration, 20 iterations are enough to avoid long running time. First of all initialize the particles randomly on the dimensions that have been provided. The range for the parameters is:

Table 5 Random Parameter Inizialitazion

Max_C	1000
Min_C	1
Max_e	1
Min_e	0.001
Max_g	10
Min_g	0.01

Then initialize the parameters contained in the particle, namely cognitive parameters (c1) and social parameters (c2). To determine the Pbest and Gbest values based on the formula (4).

Tabel 6 Example PSO Parameter

Iteration	Position			Velocity		
	x _C	x _Y	x _s	v _C	v _Y	v _s
1A	1	1	1	1	1	1
1A	1	0	3	1	1	1
1A	1	1	5	1	1	5

For example in Table 6, the initial velocity value for the first particle will be used. Assuming the value W = 0.729, C1 = 1.49445, C2 = 1.49445. rand() is 0.25 and 0.75 pbest particle 1 is (1-1-5), and gbest particle 1 is (1-1-3), then the update speed value for the first row is:

$$vC = w * vC + c1 * (p_best - x[]) * rand1 + c2 * (g_best - x[]) * rand2$$

$$= 1 * 0.729 + 1.49445 * 0.25 * (1 - 1) + 1.49445 * 0.75 * (1 - 1) = 0.729$$

$$vY = w * vY + c1 * (p_best - x[]) * rand1 + c2 * (g_best - x[]) * rand2$$

$$= 1 * 0.729 + 1.49445 * 0.25 * (1 - 1) + 1.49445 * 0.75 * (1 - 1) = 0.729$$

$$vE = w * vE + c1 * (p_best - x[]) * rand1 + c2 * (g_best - x[]) * rand2$$

$$= 1 * 0.729 + 1.49445 * 0.25 * (5 - 1) + 1.49445 * 0.75 * (3 - 1) = 44.465125$$

The update value for the second row particle is:

$$vC = w * vC + c1 * (p_best - x[]) * rand1 + c2 * (g_best - x[]) * rand2$$

$$= 1 * 0.729 + 1.49445 * 0.25 * (1 - 1) + 1.49445 * 0.75 * (1 - 1) = 0.729$$

$$vY = w * vY + c1 * (p_best - x[]) * rand1 + c2 * (g_best - x[]) * rand2$$

$$= 1 * 0.729 + 1.49445 * 0.25 * (1 - 0) + 1.49445 * 0.75 * (1 - 0) = 2.22345$$

$$vE = w * vE + c1 * (p_best - x[]) * rand1 + c2 * (g_best - x[]) * rand2$$

$$= 1 * 0.729 + 1.49445 * 0.25 * (5 - 3) + 1.49445 * 0.75 * (3 - 3) = 1.476225$$

The update value for the third row particle is:
 $vC = w * vC + c1 * (p_best - x[]) * rand1 + c2 * (g_best - x[]) * rand2$
 $= 1 * 0.729 + 1.49445 * 0.25 * (1 - 1) + 1.49445 * 0.75 * (1 - 1) = 0.729$

$v\gamma = w * v\gamma + c1 * (p_best - x[]) * rand1 + c2 * (g_best - x[]) * rand2$
 $= 1 * 0.729 + 1.49445 * 0.25 * (1 - 1) + 1.49445 * 0.75 * (1 - 1) = 0.729$

$v\epsilon = w * v\epsilon + c1 * (p_best - x[]) * rand1 + c2 * (g_best - x[]) * rand2$
 $= 1 * 0.729 + 1.49445 * 0.25 * (5 - 5) + 1.49445 * 0.75 * (3 - 5) = 1.403325$

Thus, the speed update values for the three boxes are:

Table 7 Velocity Update

Iteration	Position			Velocity			New Velocity		
	xC	xY	xs	vC	vY	vs	vC	vY	vs
1A	1	1	1	1	1	1	0.729	0.729	4.465125
1A	1	0	3	1	1	1	0.729	2.22345	1.476225
1A	1	1	5	1	1	5	0.729	0.729	1.403325

The position shift is calculated by the formula (5):

$$XNew [] = xold [] + v []$$

By using the conditions in table 7, the results of the position update are as follows:

Table 8.Update All Parameter

Iteration	Position			New Velocity			New Position		
	xC	xY	xs	vC	vY	vs	xC	xY	xs
1A	1	1	1	0.729	0.729	4.465125	1.729	1.729	5.465125
1A	1	0	3	0.729	2.22345	1.476225	1.729	2.22345	4.476225
1A	1	1	5	0.729	0.729	1.403325	1.729	1.729	6.403325

After getting a new position, the next step is to calculate the error, then it will be checked whether the error value is better than the best error value for the particle. Then do a comparison with the global error value. If the new error value is better then update the best position and error values with these two values. The process will be repeated as many particles are available.

b. Support Vector Regression

Support Vector Regression is the result of a modification of the Support Vector Maching method used to solve prediction problems [8]. In general, the concept of linear function regression can be written as follows:

$$f(x) = \langle w, x \rangle + b \quad w \in x, b \in \mathbb{R} \quad (6)$$

Where $\langle ., . \rangle$ results from the dot product in X. Function 2.1 is a flatness function to find the smallest value of w. This value can be found by minimizing the form $\|w\|^2$ This problem can be solved by *convex optimization* described in formula 7.

$$\text{Minimize} \quad \frac{1}{2} \|w\|^2 \quad (7)$$

$$\text{Subject to : } y_i - \langle w, x \rangle - b \leq \epsilon$$

$$\langle w, x \rangle + b - y_i \leq \epsilon$$

There is a condition where the error value exceeds the ϵ threshold. In this condition a soft margin is needed or it can be called a slack variable ξ_i, ξ_i^* . So the equation changes as in formula 3.

$$\text{Minimize} \quad \frac{1}{2} \|w\|^2 + C \sum_{i=0}^l (\xi_i + \xi_i^*)$$

$$\text{Subject to : } y_i - \langle w, x \rangle - b \leq \epsilon + \xi_i$$

$$\langle w, x \rangle + b - y_i \leq \epsilon + \xi_i^*$$

$$(\xi_i + \xi_i^*) \geq 0 \quad (8)$$

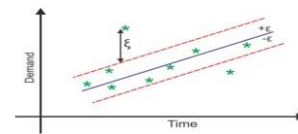


Figure 3 Support Vector Regression(SVR)

The constant $C > 0$ determines how much the error deviation from the limit ϵ can be tolerated. The above formula is a *Convex Linear Programming* NLP Optimization Problem that serves to minimize the quadratic function to be converted into a constraint. These constraints can be solved by the *Lagrange Multiplier* equation. The process of deriving the formula is very long and complicated. After going through the mathematical stages, a new equation is obtained with the following formula.

$$f(x) = \sum_{i=1}^l (\alpha_i - \alpha_i^*) \cdot \langle x_i, x \rangle + b \quad (9)$$

The function $K(x^t, x_t)$ is a Kernel . In this study, the Radial Function Basis (RBF) kernel is used which is the kernel used in the SVR calculation (Levis & Papageorgiou, 2005). Here's the formula for the RBF kernel.

$$k(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2) \quad (10)$$

The design of this algorithm uses the Support Vector Regression (SVR) method and uses the RBF kernel. The system design for the distribution of Training Data and Testing Data can be seen in the following table:

Table 9 Persentase Testing

Training	Testing
60%	40%
70%	30%
80%	20%
90%	10%

Then there is the selection of parameters from the SVR such as the values of $c, \gamma,$ and ϵ that have been determined then this test will calculate the level of accuracy of the predicted value that has been obtained and the error value. The steps taken in this research are generally divided into three main activities, namely: determining parameters using the PSO algorithm, modeling Data Training, and testing models. The results of the Data Testing will determine the results of the predicted value. The predicted value will be analyzed by regression performance, namely by calculating the error value (MAE, MSE, and RMSE). The work in outline can be seen as Figure 3 Block Diagram of Algorithm Design.

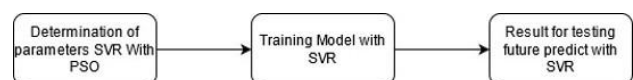


Figure 4 Block Diagram Algorithm

The first stage is Data Training modeling, shown in Figure 3 with electricity data for the P Telkom University building. This initial stage aims for the SVR method to study the existing dataset model. This stage is accompanied by the selection of a kernel type, namely the RBF kernel.



Figure 5 Flowchart Training Model

The second stage is to find the optimal parameter values. Selection of parameter values using the PSO Algorithm method as a complement to the shortcomings of the SVR method in each hyperplane. The second stage can be seen in Figure 4 Flow Chart of the PSO Algorithm.

The third stage, namely Testing, is shown in Figure 5, with the previous training stage modeling. In this process, the testing data is processed. The smaller the error rate or the close accuracy of the regression forecast function, the more accurate the prediction results. The accuracy value can be calculated by calculating the error value of the MAE, MSE and RMSE indicators.

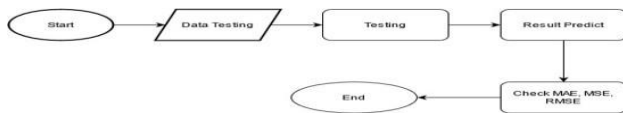


Figure 6 Flowchart Testing Result

4. Result Evaluation

The accuracy of a prediction system needs to be reviewed in order to determine its performance, because the predictor system will not escape errors in its performance. To examine the accuracy of the forecasting model, it can be compared between the forecasting results from the model and the actual data available. If a model produces a minimum error value, the forecasting method is considered a good method for forecasting. Some calculations that have been commonly used to monitor prediction systems to ensure whether a prediction system is running well include:

- a) Mean Absolute Error (MAE)

$$MAE = \frac{\sum_{i=1}^n |y_i - x_i|}{n} \tag{11}$$

y_i = prediction

x_i = true value

n = total number of data points

- b) Mean Square Error (MSE)

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 \tag{12}$$

n = total number of data points

Y_i = observed values

\hat{Y}_i = predicted values

- c) Root Mean Square Error (RMSE)

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (Y_i - \hat{Y}_i)^2}{n}} \tag{13}$$

IV. RESULT AND ANALYSIS

In this section, testing of applications that have been made in accordance with the results of application design and implementation is carried out. Then an analysis of the test results is carried out so that the success of the system can be known.

In this research conducted two types of system testing, namely alpha testing and beta testing. Alpha testing is done by observing the execution results through test data and checking the functionality of the website. While beta testing is done by taking some data for research. After carrying out these two types of tests, a comparison of the prediction results will then be carried out.

1. Alpha Testing

Alpha testing is a type of testing which is done to identify all possible problems/bugs before releasing the product to users. The test is said to have been successful if the results of the output are in accordance with the input control. This test is carried out from the process of starting the application, namely starting the application, the login / registration process, the prediction process, uploading the dataset, and logging out and then getting the accuracy results from the website that has been created.

$$Acuration = \frac{\text{Testing Succeed}}{\text{Total Testing}} \times 100 \% \tag{14}$$

$$Acuration = \frac{23}{23} \times 100 \% = 100 \%$$

Based on the above calculations it can be concluded that the accuracy of the website is 100%

2. Testing SVR Parameter

Testing the parameters of the SVR method is carried out using several stages. The first stage will be testing the data partition, after finding the best data partition, SVR parameters will be tested without using PSO optimization. At the end of the test, we will try to find the best SVR parameters by using PSO optimization.

a. Testing Split data

System testing for the distribution of Data Training and Data Testing begins with the default SVR parameters, namely $C = 1$, $\gamma = 0.125$, $\epsilon = 0.1$. The test results can be seen in the table below.

Table 10 Testing Partition



It was found that the partition that has the smallest error value is a comparison of 90% training and 10% testing.

b. Testing Without PSO

1) Testing C parameter

Table 11 Testing C Parameter

C	Gamma	Epsilon	MAE	MSE	RMSE
1	0.125	0.1	0.0207081	0.0111719	0.105697
10	0.125	0.1	0.0209594	0.0111769	0.105721
100	0.125	0.1	0.0208687	0.0111743	0.105709
1000	0.125	0.1	0.020903	0.0111744	0.105709

Dari pengujian parameter C ditemukan bahwa parameter C terbaik pada nilai C = 100 dengan nilai MAE = 0.0208687; MSE = 0.0111743; RMSE 0.105709.

2) Testing Gamma Parameter

Gamma parameter testing will be tested with parameters 0.001 , 0.01 , 0.125 , 1 , 10 , for parameter C the best parameter from the previous test will be used, namely C = 100, and the value of epsilon = 0.1.

Table 12 Testing Gamma Parameter

Gamma	C	Epsilon	MAE	MSE	RMSE
0.01	100	0.1	0.0206812	0.0111722	0.105699
0.125	100	0.1	0.0208687	0.0111743	0.105709
1	100	0.1	0.0207086	0.0111697	0.105687
10	100	0.1	0.0170797	0.00623928	0.0789891

From the test results it was found that the best gamma parameter is gamma = 10 with an error value of MAE = 0.0170797 , MSE = 0.00623928 , 0.0789891

3) Testing Epsilon Parameter

For testing the Epsilon parameter, the epsilon parameter will be used with a value of 0.001 , 0.01 , 0.1 , 1 , 10

Table 13 Testing Epsilon Parameter

Epsilon	C	Gamma	MAE	MSE	RMSE
0.001	100	10	0.0084756	0.00622745	0.0789142
0.01	100	10	0.0170797	0.00623928	0.0789891
0.1	100	10	0.106465	0.015358	0.123927
1	100	10	0.994745	0.995136	0.997565

c. Testing with PSO Parameter

In testing the PSO parameters, the Parameter Number of particles ranging from 20 to 80 and, Iterations of 20, Dimensions of 3, and Inertia of 1.

Table 14 Result Best Parameter With PSO

Partikel	C	Gamma	Epsilon	MAE	MSE	RMSE
20	1	8.9045	0.001	0.0083048	0.0060225	0.0776048
30	1	8.9938	0.001	0.0083147	0.00602239	0.0776041
40	1	8.9883	0.001	0.00829921	0.00602241	0.0776042
50	1	9.0138	0.001	0.00831927	0.00602246	0.077604
60	1	8.9878	0.001	0.00830563	0.0060224	0.0776041
70	1	8.9918	0.001	0.0083092	0.00602239	0.077604
80	1	8.9889	0.001	0.00830103	0.00602241	0.0776042

d. Parameter Analysis

Analysis of the SVR method obtained the average value of the resulting parameter for the value of C = 1, the value of epsilon = 0.01 and the value of gamma = 8.9. In the test results, it can be seen that the value of C means ignoring the noise value. The epsilon value has the meaning of

determining how many errors the SVR receives by defining a large epsilon range where there is no penalty associated with the true value of the point. Gamma value is a coefficient in a kernel, the kernel used in this study is RBF (Radial Basis Function).The results of the Gamma value test are close to 0 which means that the results of the Gamma value are close to the original data but can cause overfitting.

Analysis of the PSO method obtained the average value of the resulting parameter for the value of C = 1; gamma=8.9 ; epsilon = 0.001. The kernel used is the same as the SVR, namely RBF (Radial Basic Function). If the value of the Test Size is increased to 20% then the MSE and MAE are getting smaller or closer to 0 which means the prediction results are close to the real value.

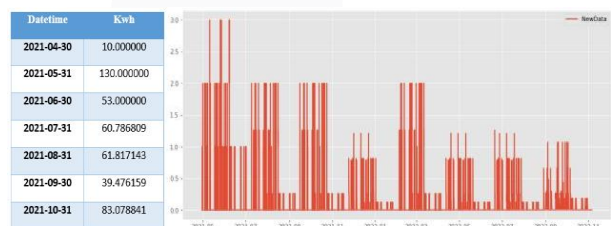
e. Performance Analysis

From the results of using PSO optimization, the SVR parameter values obtained are C = 1, Gamma = 8.9, Epsilon = 0.001; produces an error value, MAE = 0.00829921; MSE = 0.00602241; RMSE = 0.0776042. The SVR method still produces a predictive value for the number of electrical prediction intervals based on the existing data model. The purpose of the MAE, MSE and RMSE values is as a measurement of predictive accuracy, because a small error value indicates an effective model performance.

f. Testing Future Predict and Price

Based on the tests carried out previously, predictions of electrical data will be carried out using the best parameters that have been searched which will then be searched for future values. Here are the test results with the best parameters:

Table 15 Result Energy Predict



In calculating the rupiah price for the P or FTE building using the social electricity class, it can be concluded, per Kwh, based on PERMEN ESDM No. 28 of 2016. Therefore, the total price each month that must be paid is in the appendix. The total kVarh in the P building is 500 kVarh which is obtained from the information of supervisor 1, namely Mrs. Casi Setianingsih S.T. M.T. Then it can be calculated as follows using the equation

$$LWBP = 1.3 \times 735 = 955,5 \quad (15)$$

$$E\text{lectricity Price Each Month} = 40 \left(\frac{401}{500} \right) \times 955,5 \quad (16)$$

$$= 30.65244$$

Table 16 Price Predict

Datetime	Kwh	Price
30/04/2021	10	Rp 306.524
31/05/2021	130	Rp 3.984.817
30/06/2021	53	Rp 1.624.579
31/07/2021	61	Rp 1.863.264
31/08/2021	62	Rp 1.894.846
30/09/2021	39	Rp 1.210.041

3. Beta Testing

Beta testing is the second stage of software testing where users try out the product. The goal of beta testing is to put the app in the hands of real users. Testing is done by distributing questionnaires filled out by respondents. The questionnaire made consisted of an assessment of the idea, appearance of the application, and application performance. The test was carried out on 59 respondents for 1 week from 26 July 2021 to 1 August 2021. There were 8 questions with a weight of 1 (Very Difficult), 2 (Difficult), 3 (Ordinary), 4 (Easy), 5 (Very Easy) . The results of the questionnaire are as follows:

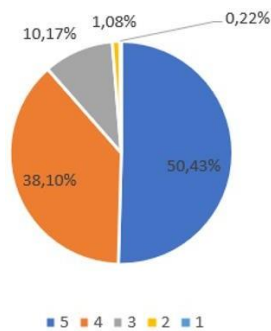


Figure 7 Result Beta Testing

Based on the graph above, it can be seen that about 50.43% of respondents chose answer 5 which is very easy. It can be concluded that the website is feasible and easy to use by users.

V. CONCLUSION

Based on the results of research, testing and analysis that have been carried out in this final project, the following conclusions can be drawn:

1. The electricity usage prediction system with SVM, and website-based electricity usage prediction media has been running well with 100% alpha test results.
2. The results of testing the parameters of the Support Vector Machine algorithm on the electricity usage prediction system, the lowest error values obtained are MAE, MSE, RMSE on the parameters of the PSO optimization results, the SVR parameter value is $C = 1$; $\text{Gamma} = 8.9$; $\text{Epsilon} = 0.001$;

produces an error value, $\text{MAE} = 0.00829921$; $\text{MSE} = 0.00602241$; $\text{RMSE} = 0.0776042$.

3. The results of testing the difference in error rates before and after the system is optimized with the PSO algorithm show a decrease in the error rate although it is not significant, this proves that the SVR algorithm can be optimized using Particle Swarm Optimization to make it easier to find the best SVR parameters without the need to waste a lot of time manually determining parameters .

VI. REFERENCES

- [1] E. Mocanu dan M. G. W. L. K. Phuong H. Nguyen, "Deep Learning for estimating building energy consumption," Elsevier, vol. 6, pp. 91-99, 2016
- [2] B. Santosa, Data mining teknik pemanfaatan data untuk keperluan bisnis., Yogyakarta: Graha Ilmu, 2007.
- [3] S. Suprianto, "Implementasi Algoritma Naive Bayes Untuk Menentukan Lokasi Strategis Dalam Membuka Usaha Menengah Ke Bawah di Kota Medan," Jurnal Sistem komputer dan Informatika, p. 126, 2020.
- [4] Kennedy, J. and Eberhart, R. 1995. Particle Swarm Optimization, IEEE International Conference on Neural Networks (Perth, Australia), IEEE Service Center, Piscataway, NJ, IV: 1942-1948.
- [5] Hasan, Rania. Particle Swarm Optimization: Method and Application. Engineering Systems Division - Massachusetts Institute of Technology. 2004.
- [6] Hossein Lotfi. 2014. Short and Mid-Term Wind Power Forecasting with ANN-PSO. ISSN: 2321-3124.
- [7] Bai Qinghai, "Analysis of Particle Swarm Optimization Algorithm" College of Computer Science and Technology Inner Mongolia University for Nationalities Tongliao 028043 China, 2010.
- [8] A. Smola and B. Schölkopf, "A tutorial on support vector regression," Stat. Comput., vol. 14, pp. 199–222, 2004.