Jurnal CoreIT, Vol.11, No.2, December 2025
ISSN 2460-738X (Print)
ISSN 2599-3321 (Online)

A Support Vector Regression Approach for Predicting the Remaining Useful Life of Turbofan Engines

Muhammad Vio Hardiansyah[1], Fitri Insani[1*], Lestari Handayani[1], Jasril[1], Suwanto Sanjaya[1]
[1]Dept. of Informatics Engineering, Universitas Islam Negeri Sultan Syarif Kasim, Indonesia
fitri.insani@uin-suska.ac.id[*], lestari.handayani@uin-suska.ac.id, jsaril@uin-suska.ac.id, suwanto.sanjaya@uin-suska.ac.id

Abstract. Turbofan engines are crucial components in the aviation and manufacturing industries, where estimating the Remaining Useful Life (RUL) has a significant impact on operational efficiency and safety. This study aims to predict the RUL of turbofan engines using the Support Vector Regression (SVR) method, a machine learning approach that has proven effective in modeling nonlinear relationships between variables. Operational data related to turbofan engines include operational parameters, sensors, and maintenance records. The initial stage of this research involves data analysis based on unit number, time, operational control, and sensor parameters. This process begins with preprocessing to initialize the initial data values, normalize, and select sensors that have stagnant values, as these sensors do not affect the machine learning system. Subsequently, regression calculations are performed to compare predicted values and actual values using the Support Vector Regression method optimized with Grid Search Optimization. In this study, testing was conducted with Parameters C [1, 10, 50, 100] and ε [1, 5, 10, 50], resulting in the best model with an RMSE error of 19.56 and MAE of 14.73.

Keywords: Grid Search Optimization, Prediction, RUL, SVR, Turbofan.

Received November 2025 / **Revised** December 2025 / **Accepted** December 2025

INTRODUCTION

C-MAPSS (Commercial Modular Aero Propulsion System Simulation) is a dataset published by NASA containing simulated flight data from various measurements. This dataset is intended for research purposes and is used in this study. Several publications and research studies have been conducted related to this dataset. The C-MAPSS dataset simulates flight data recordings from various measurements that occur in aircraft engines [1]. This dataset consists of four datasets, with each dataset further divided into training data and test data. Each turbofan engine has 21 sensors that collect data for measuring temperature, speed, pressure, and ratios. The first column of the dataset is the unit number and operational time [2]. To maintain engine health, operational maintenance is required. This drives the need for prognostic processes to predict the likelihood of failures and estimate the remaining effective operational time of components before failure occurs (Remaining Useful Life/RUL) [3]. By understanding RUL, we can measure the lifespan of a machine. Predictive maintenance model is estimating the time of equipment failure to schedule maintenance times [4].

With machine maintenance and care using data analysis to detect damage to machine components, damage can be predicted and repaired before it occurs in the machine. Therefore, an in-depth study and analysis are very necessary. Previously, a case study of this CMAPSS data was conducted using the Elastic Weight Consolidation (EWC) method. It was found that EWC performance can be affected when the tasks faced have different complexities. The results of this study indicate that the proposed approach is able to compete with other approaches in predicting turbofan engine degradation [5]. In addition, other researchers used the Convolutional Neural Network (CNN) and Transformer Encoder methods. The results show that this combined model demonstrates excellent performance and is comprehensively superior [6].

This research uses Support Vector Regression (SVR). Because SVR in previous research was able to predict cases of bearing damage detection using the SVR method with the best parameters resulting in RMSE for training data of 4.5785 and RMSE for test data of 9.6796 [7]. In addition, there are also researchers using the SVR method with K-fold error predicting COVID-19 patients, obtaining the most optimal model for active COVID-19 cases of 0.000547 and the largest error model on the death attribute of 0.575177 [8]. The

DOI: 10.24014/coreit.v11i2.38532

SVR method was also used by Isnaeni R. (2022) to predict the inflation rate in Indonesia using the SVR method with RBF kernel, resulting in an RMSE value of 0.0020 [9].

Based on previous studies, the use of the SVR method obtains good accuracy values and can be used to predict Remaining Useful Life (RUL). The use of SVR has been successfully applied to several problems in time series prediction[10]. So that time series data prediction can be carried out. On this turbofan engine usingan artificial intelligence method, namely Support Vector Regression (SVR). The Support Vector Regression method has been widely used to assist researchers in prediction or forecasting cases with a fairly low error rate [11]. SVR, which is part of Support Vector Machine (SVM) introduced by Vapnik in 1995, is used for regression and prediction cases.

The SVR algorithm can produce accurate forecast values because it has the ability to overcome overfitting problems [9]. Overfitting is a condition where the model predicts training data with almost perfect accuracy, but its performance decreases significantly when applied to test data[10].

METHODS

The research methodology is first carried out by identifying the problem, after that the preprocessing process reaches the stage normalize the data until the data is ready for the modeling process.

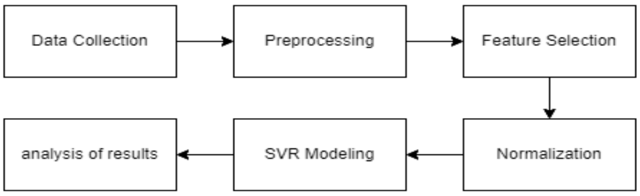

Figure 1. Research stages

C-MAPSS Dataset

C-MAPSS is a tool encoded in the MATLAB-Simulink environment to simulate a 90,000lb thrust class engine model [12]. By using a number of editable input parameters, it allows for determining operational profiles, closed-loop control, environmental conditions (various altitudes and temperatures), in addition to provisions for modifying several efficiency parameters to simulate various degradations in different parts of the engine system [13]. To model uncertainties in meter readings during operation, additionally, each sensor has its own initial wear level and manufacturing variance. This engine contains 21 sensors, operating conditions, and other information. Since each time series represents a different engine, the data from the C-MAPSS collection can be considered to come from the same engine [2].

Table 1. CMAPSS Dataset

Dataset	Train Trejectories	Test Trejectories	Conditions	Fault Mode
FD001	100	100	One (Sea level)	One (HPC)
FD002	260	259	Six	One (HPC)
FD003	100	100	One (Sea level)	Two (HPC, Fan)
FD004	248	249	Six	Two (HPC, Fan)

Table 2. Variabel CMAPSS Dataset

Index	Dataset Description
1	Unit Number
2	Time (In Cycle)
3	Operational Setting 1
4	Operational Setting 2
5	Operational Setting 3
6	Sensor Measurement 1
…	
26	Sensor Measurement 21

Support Vector Regression

Support Vector Regression is part of the supervised learning algorithm, used to predict the value of continuous variables. Like the SVM concept, the SVR method also seeks the best hyperplane in the form of a regression function by minimizing errors and maximizing margins. The goal of SVR is to find a function f(x) as a hyperplane (separating line) in the form of a regression function that fits all input data, by making the error as small as possible [14].

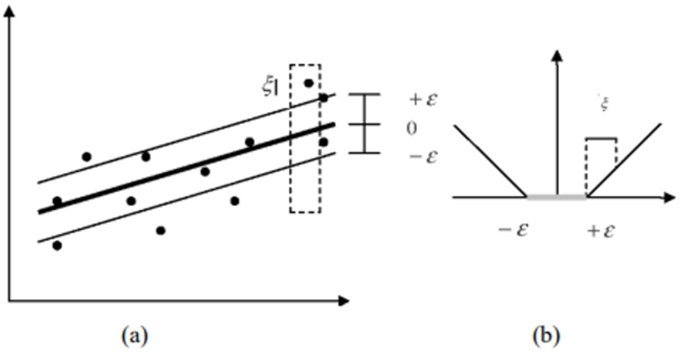

Figure 2. Concept of Support Vector Regression

In Figure 2(a), the thick black line represents a hyperplane, while the two lines flanking it are soft margins. The distance between the hyperplane and the soft margin is ε, and the points that lie between +ε and -ε are called support vectors. However, for points that exceed the soft margin, slack variables ξ are needed. The basic idea of using the SVR method is to find a regression function that fits the given training data, consisting of n sets of data (x_i, y_i) with xi ∈ $\mathbb{R}^d$ as the input vector of the i-th data, where i = 1, 2, ..., n, and y_i is the corresponding target value. The SVR method aims to produce a model that can predict the target value y for new, unseen input x, utilizing the hyperplane concept learned from the training data. In Support Vector Regression, if we want to find a function f(x) that has the largest deviation ε from the target yi for training data, it can be seen in the formula 1.

$$f(x) = \langle w.x \rangle + b \qquad (1)$$

$f(x)$ is the function of SVR (Support Vector Regression x is the input vector w is the weight vector of dimension l and b is the bias.

$$L_\varepsilon(y_1, f(x_i)) = \begin{cases} 0;\ y_i - f(x_i) \le \varepsilon \\ |yi - f(xi)| - \varepsilon; \end{cases} \qquad (2)$$

Formula 2 is the equation. In general, a linear function can be expressed as ⟨w, x⟩ + b, where ⟨.⟩ is the dot product in x. To maximize the hyperplane, it is necessary to minimize the Loss Function.

$$R(f(x)) = \frac{1}{2}\, \|w\|^2 + \frac{c}{n} \Sigma_{i=1}^{n} L_\varepsilon(y_1, f(x_i)) \qquad (3)$$

Using the ε-insensitive loss function Lε, and with parameters C and ε. Using dual formulation through Lagrange multipliers, Support Vector Regression (SVR) can be extended to include non-linear functions. In this context, the optimization problem is formulated only in terms of Lagrange multipliers α_i and α_i*. This is possible because the kernel function (x_i, x_j) returns the dot product between pairs of data in high-dimensional space without explicitly mapping the data to that space [15].

Prepsocessing

Preprocessing is a technique to manipulate data collected from various sources so that it's ready for use in subsequent processes. This process is important to address issues such as missing values, redundant data, outliers, or data formats that are incompatible with the system. The presence of these problems can interfere with the final results of the output produced [16].

The first preprocessing step is Min-Max Normalization, where each value in the data feature is subtracted by the minimum value of that feature, then the result is divided by the difference between the maximum and minimum values of that feature [17]. Where x_{new} is the normalized value of the attribute, x_{min} is the minimum value of the attribute, and x_{max} is the maximum value of the attribute.

$$x_{new} = \frac{x_{old} - x_{min}}{x_{max} - x_{min}} \qquad (4)$$

Next, Standard Scaler normalization is a preprocessing method where each feature in the sample will be transformed in such a way that the feature has a mean of zero and a variance of one. This process is done by subtracting the mean from each feature and then dividing it by the standard deviation of that feature [18]. Where $\bar{x}$ is the sample mean and σ is the standard deviation.

$$Z = \frac{x_i - \bar{x}}{\sigma} \qquad (5)$$

Grid Search Optimization

This algorithm works by dividing the range of parameters to be optimized into a grid and testing all combinations of grid points to obtain the optimal parameters. In its performance, this grid-search algorithm must be guided by several performance metrics measured by cross-validation on the training data. Therefore, it is recommended to try several variations of pairs on the Support Vector Regression hyperplane to find the most optimal model [19].

Model selection is based on the performance results of the models that have been created. The evaluations used to measure the quality of model predictions include Root Mean Square Error (RMSE) and Mean Absolute Error (MAE).

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)} \qquad (6)$$

The RMSE calculation involves squaring the differences between predicted and actual values, taking the average of these squared differences, and then finding the square root of this average. This gives us a measure of the typical magnitude of prediction errors.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|Y_i - \hat{Y}_i\right| \qquad (7)$$

The MAE is calculated by taking the absolute differences between predicted and actual values, summing these differences, and then dividing by the number of observations. This gives us an average of the absolute errors, providing a straightforward measure of prediction accuracy.

Both RMSE and MAE are in the same units as the original data, making them interpretable in the context of the problem. However, RMSE tends to penalize large errors more heavily due to the squaring step, while MAE treats all sizes of errors linearly. The choice between these metrics often depends on the specific requirements of the prediction task and the nature of the errors that are most important to minimize in your particular application.

RESULT AND DISCUSSION

Data is a crucial element in research. Without data, research cannot be conducted. Data processing is necessary to generate useful information to support decision-making or develop theories. Therefore, the first step in research is to collect the required data. After the C-MAPSS dataset is collected, the research focus shifts to the stages of data analysis and interpretation as an initial phase in developing predictive models. Data from the files RUL_FD001.txt, test_FD001.txt, and train_FD001.txt have been prepared for further analysis.

In the Train, Test, and RUL FD001 datasets, there are several columns that will be the focus of analysis. The details of Train FD001 data contain information about the unit, operating system, and measured sensor data. During the initial analysis, it was noticed that the amount of sensor data in columns 5, 9, 10, and others have the same values. Researchers will check each column to see if there are significant changes in degradation. Columns that do not show much change in degradation will be removed during the pre-processing and data cleaning stages, as well as in subsequent stages.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
0	1	1	-0.0007	-0.0004	100.0	518.67	641.82	1589.70	1400.60	14.62	21.61	554.36	2388.06	9046.19	1.3	47.47	521.66	2388.02	8138.62	8.4195	0.03	392	2388
1	1	2	0.0019	-0.0003	100.0	518.67	642.15	1591.82	1403.14	14.62	21.61	553.75	2388.04	9044.07	1.3	47.49	522.28	2388.07	8131.49	8.4318	0.03	392	2388
2	1	3	-0.0043	0.0003	100.0	518.67	642.35	1587.99	1404.20	14.62	21.61	554.26	2388.08	9052.94	1.3	47.27	522.42	2388.03	8133.23	8.4178	0.03	390	2388
3	1	4	0.0007	0.0000	100.0	518.67	642.35	1582.79	1401.87	14.62	21.61	554.45	2388.11	9049.48	1.3	47.13	522.86	2388.08	8133.83	8.3682	0.03	392	2388
4	1	5	-0.0019	-0.0002	100.0	518.67	642.37	1582.85	1406.22	14.62	21.61	554.00	2388.06	9055.15	1.3	47.28	522.19	2388.04	8133.80	8.4294	0.03	393	2388

Figure 3. Train FD001 Data

In Figure 3, which displays the Train FD001 data, initial analysis results show that there is no significant variation indicating data decline or degradation yet. The range of sensor values at the beginning of the cycle seems relatively consistent for each type of sensor. Nevertheless, to detect degradation or patterns of decline more accurately from this data, a more in-depth analysis is required in the subsequent stages.

Analysis of train and test data

```
array([  1,   2,   3,   4,   5,   6,   7,   8,   9,  10,  11,  12,  13,
        14,  15,  16,  17,  18,  19,  20,  21,  22,  23,  24,  25,  26,
        27,  28,  29,  30,  31,  32,  33,  34,  35,  36,  37,  38,  39,
        40,  41,  42,  43,  44,  45,  46,  47,  48,  49,  50,  51,  52,
        53,  54,  55,  56,  57,  58,  59,  60,  61,  62,  63,  64,  65,
        66,  67,  68,  69,  70,  71,  72,  73,  74,  75,  76,  77,  78,
        79,  80,  81,  82,  83,  84,  85,  86,  87,  88,  89,  90,  91,
        92,  93,  94,  95,  96,  97,  98,  99, 100], dtype=int64)
```

Figure 4. Analysis of 100 Engine Units in Train Data

The analysis results in Figure 4 show that the training dataset includes 100 engine units, ranging from unit 1 to 100. The involvement of a large number of engine units in this analysis provides a comprehensive representation of the operational conditions of the engines. This information has significance in developing accurate and reliable prediction models. With the variety of engine units recorded in the dataset, the analysis results can be more representative and relevant in supporting decision-making related to engine maintenance and condition monitoring.

```
array([192, 287, 179, 189, 269, 188, 259, 150, 201, 222, 240, 170, 163,
       180, 207, 209, 276, 195, 158, 234, 195, 202, 168, 147, 230, 199,
       156, 165, 163, 194, 234, 191, 200, 195, 181, 158, 170, 194, 128,
       188, 216, 196, 207, 192, 158, 256, 214, 231, 215, 198, 213, 213,
       195, 257, 193, 275, 137, 147, 231, 172, 185, 180, 174, 283, 153,
       202, 313, 199, 362, 137, 208, 213, 213, 166, 229, 210, 154, 231,
       199, 185, 240, 214, 293, 267, 188, 278, 178, 213, 217, 154, 135,
       341, 155, 258, 283, 336, 202, 156, 185, 200], dtype=int64)
```

Figure 5. Analysis of 128-362 Cycles Before Failure in Train Data

Figure 5 displays the analysis results regarding the number of cycles each engine goes through before experiencing failure. This data illustrates the variation in the number of cycles before failure, ranging from 128 to 362 cycles. This information provides important insights into the relative operational life of each engine before failure occurs. By understanding the number of cycles required before failure occurs, maintenance and care planning can be optimized to reduce the risk of failure and improve model performance.

The column labeling process for the C-MAPSS dataset begins with naming each feature in the dataset. This dataset consists of three operational setting columns (os1, os2, os3) which indicate the operational settings of the observed unit, and 21 sensor columns (s1 to s21) which record sensor signal data from that unit. The initial step in this process is to create an empty list to store the new column names. We will add the first two columns, 'unit' and 'time', which represent the observed unit number and observation time. After that, we iterate from 1 to 3 to add column names for the operational setting columns with the format 'os' followed by the iteration number. The same is done for the sensor columns, iterating from 1 to 21 and adding column names with the format 's' followed by the iteration number.

Feature selection becomes an important part of data processing because we can see several sensor sources that show stagnant or constant trends, which means the sensor values tend to stay within a certain range without significant changes over time. This can have a quite important impact on the modeling process that will be carried out.

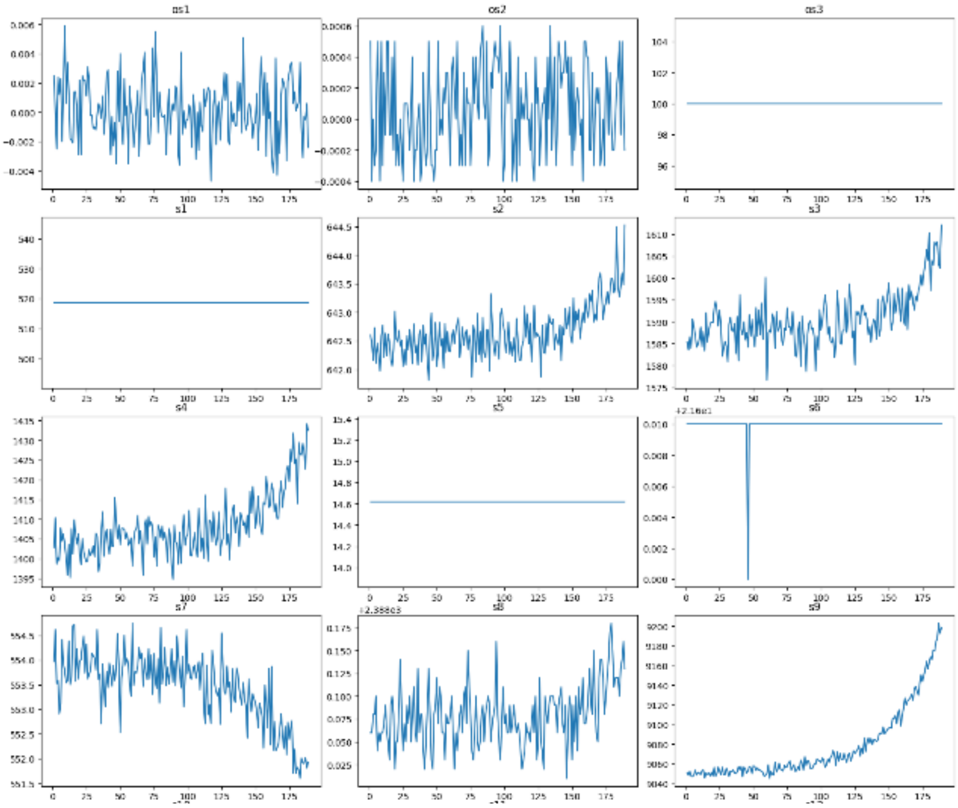

Figure 6. Data Visualization

From the graph in Figure 6, we can see that Operating Condition 3, as well as sensors 1, 5, 6, 10, 16, 18, 19, and 21 show a stagnant tendency. This indicates that the data from these sensors do not provide useful or relevant information for the modeling process. This stagnation tendency can lead to model inaccuracies and less reliable interpretations. It is evident from the visualization above that there are several sensors that need to be removed because they have no influence on the prediction process due to their stagnant values. Figure 7 show the results after dropping stagnant data from train and test sets. Sensors like os3, s1, s5, s6, s10, s16, s18, s19, and s21 are considered to not contribute significantly to the model because their data distribution is inconsistent or has low variation.

	unit	time	os1	os2	s2	s3	s4	s7	s8	s9	s11	s12	s13	s14	s15	s17	s20	s21	rul
0	1	1	-0.0007	-0.0004	641.82	1589.70	1400.60	554.36	2388.06	9046.19	47.47	521.66	2388.02	8138.62	8.4195	392	39.06	23.4190	191
1	1	2	0.0019	-0.0003	642.15	1591.82	1403.14	553.75	2388.04	9044.07	47.49	522.28	2388.07	8131.49	8.4318	392	39.00	23.4236	190
2	1	3	-0.0043	0.0003	642.35	1587.99	1404.20	554.26	2388.08	9052.94	47.27	522.42	2388.03	8133.23	8.4178	390	38.95	23.3442	189
3	1	4	0.0007	0.0000	642.35	1582.79	1401.87	554.45	2388.11	9049.48	47.13	522.86	2388.08	8133.83	8.3682	392	38.88	23.3739	188
4	1	5	-0.0019	-0.0002	642.37	1582.85	1406.22	554.00	2388.06	9055.15	47.28	522.19	2388.04	8133.80	8.4294	393	38.90	23.4044	187

	unit	time	os1	os2	s2	s3	s4	s7	s8	s9	s11	s12	s13	s14	s15	s17	s20	s21	rul
0	1	1	0.0023	0.0003	643.02	1585.29	1398.21	553.90	2388.04	9050.17	47.20	521.72	2388.03	8125.55	8.4052	392	38.86	23.3735	142
1	1	2	-0.0027	-0.0003	641.71	1588.45	1395.42	554.85	2388.01	9054.42	47.50	522.16	2388.06	8139.62	8.3803	393	39.02	23.3916	141
2	1	3	0.0003	0.0001	642.46	1586.94	1401.34	554.11	2388.05	9056.96	47.50	521.97	2388.03	8130.10	8.4441	393	39.08	23.4166	140
3	1	4	0.0042	0.0000	642.44	1584.12	1406.42	554.07	2388.03	9045.29	47.28	521.38	2388.05	8132.90	8.3917	391	39.00	23.3737	139
4	1	5	0.0014	0.0000	642.51	1587.19	1401.92	554.16	2388.01	9044.55	47.31	522.15	2388.03	8129.54	8.4031	390	38.99	23.4130	138

Figure 7. Results after dropping stagnant data from train and test sets

Figure 8 show the normalization result using the Min-Max Scaling method, which aims to transform the values in each feature to a range between 0 and 1. Figure 9 show Normalization Result with Standard Scaler. The purpose of this process is to ensure that all features have a uniform scale, so that no feature dominates others in its influence on the model.

os1	os2	s2	s3	s4	s7	s8	s9	s11	s12	s13	s14	s15	s17	s20	s21	rul
0.459770	0.166667	0.183735	0.406802	0.309757	0.726248	0.242424	0.109755	0.369048	0.633262	0.205882	0.199608	0.363986	0.333333	0.713178	0.724662	191
0.609195	0.250000	0.283133	0.453019	0.352633	0.628019	0.212121	0.100242	0.380952	0.765458	0.279412	0.162813	0.411312	0.333333	0.666667	0.731014	190
0.252874	0.750000	0.343373	0.369523	0.370527	0.710145	0.272727	0.140043	0.250000	0.795309	0.220588	0.171793	0.357445	0.166667	0.627907	0.621375	189
0.540230	0.500000	0.343373	0.256159	0.331195	0.740741	0.318182	0.124518	0.166667	0.889126	0.294118	0.174889	0.166603	0.333333	0.573643	0.662386	188
0.390805	0.333333	0.349398	0.257467	0.404625	0.668277	0.242424	0.149960	0.255952	0.746269	0.235294	0.174734	0.402078	0.416667	0.589147	0.704502	187

os1	os2	s2	s3	s4	s7	s8	s9	s11	s12	s13	s14	s15	s17	s20	s21	rul
0.632184	0.750000	0.545181	0.310661	0.269413	0.652174	0.212121	0.127614	0.208333	0.646055	0.220588	0.132160	0.308965	0.333333	0.558140	0.661834	142
0.344828	0.250000	0.150602	0.379551	0.222316	0.805153	0.166667	0.146684	0.386905	0.739872	0.264706	0.204768	0.213159	0.416667	0.682171	0.686827	141
0.517241	0.583333	0.376506	0.346632	0.322248	0.685990	0.227273	0.158081	0.386905	0.699360	0.220588	0.155640	0.458638	0.416667	0.728682	0.721348	140
0.741379	0.500000	0.370482	0.285154	0.408001	0.679549	0.196970	0.105717	0.255952	0.573561	0.250000	0.170090	0.257022	0.250000	0.666667	0.662110	139
0.580460	0.500000	0.391566	0.352082	0.332039	0.694042	0.166667	0.102396	0.273810	0.737740	0.220588	0.152751	0.300885	0.166667	0.658915	0.716377	138

Figure 8. Normalization Results with Min-Max

	unit	0s1	0s2	s2	s3	s4	s7	s8	s9	s11	s12	s13	s14	s15	s17	s20
0	1.0	-0.315980	-1.372953	-1.721725	-0.134255	-0.925936	1.121141	-0.516338	-0.862813	-0.266467	0.334262	-1.058890	-0.269071	-0.603816	-0.781710	1.348493
1	1.0	0.872722	-1.031720	-1.061780	0.211528	-0.643726	0.431930	-0.798093	-0.958818	-0.191583	1.174899	-0.363646	-0.642845	-0.275852	-0.781710	1.016528
2	1.0	-1.961874	1.015677	-0.661813	-0.413166	-0.525953	1.008155	-0.234584	-0.557139	-1.015303	1.364721	-0.919841	-0.551629	-0.649144	-2.073094	0.739891
3	1.0	0.324090	-0.008022	-0.661813	-1.261314	-0.784831	1.222827	0.188048	-0.713826	-1.539489	1.961302	-0.224597	-0.520176	-1.971665	-0.781710	0.352598
4	1.0	-0.864611	-0.690488	-0.621816	-1.251528	-0.301518	0.714393	-0.516338	-0.457059	-0.977861	1.052871	-0.780793	-0.521748	-0.339845	-0.136018	0.463253

Figure 9. Normalization Result with Standard Scaler

Application of support vector regression algorithm

Parameter determination in the Support Vector Regression (SVR) method is very important as it can affect the performance and ability of the model to make accurate predictions. In the context of using the SVR method, parameter determination involves selecting optimal values for certain parameters that can influence the strength and flexibility of the model. In the process of determining SVR parameters, there are several main parameters that need to be considered, including the C parameter, epsilon, and kernel. In this case, the researcher uses the Radial Basis Function (RBF) Kernel due to its flexibility in handling complex data. With the C parameter defined as [1, 10, 50, 100] and epsilon as [1, 5, 10, 50].

Grid search can be used to produce the best parameter combination by trying all possible combinations of values for each specified parameter and evaluating performance using cross-validation techniques to determine the optimal parameter combination. From the results of this Grid Search test, which produces various combinations of several parameters, the results are obtained as shown in Table 3.

Table 3. Test scenarios

Normalization	**Parameter**		**Value**		**Normalization**	**Parameter**		**Value**	
	C	**ε**	**RMSE**	**MSE**		**C**	**ε**	**RMSE**	**MSE**
Min-Max	1	1	49,04	38,44	Standard Scaler	1	1	20,79	15,23
Min-Max	1	5	49,05	38,44	Standard Scaler	1	5	20,52	15,11
Min-Max	1	10	49,04	38,43	Standard Scaler	1	10	20,00	14,88
Min-Max	1	50	41,44	36,76	Standard Scaler	1	50	34,07	30,30
Min-Max	10	1	44,26	34,42	Standard Scaler	10	1	20,75	15,23
Min-Max	10	5	44,29	34,44	Standard Scaler	10	5	20,33	15,07
Min-Max	10	10	44,00	34,27	Standard Scaler	10	10	19,73	14,78
Min-Max	10	50	40,63	36,17	Standard Scaler	10	50	33,63	29,87
Min-Max	50	1	34,87	28,66	Standard Scaler	50	1	20,73	15,35
Min-Max	50	5	34,79	28,59	Standard Scaler	50	5	20,26	15,13
Min-Max	50	10	34,67	28,52	Standard Scaler	50	10	19,64	14,76
Min-Max	50	50	39,12	34,85	Standard Scaler	50	50	33,93	30,18
Min-Max	100	1	33,20	27,32	Standard Scaler	100	1	20,67	15,34
Min-Max	100	5	33,01	27,07	Standard Scaler	100	5	20,25	15,19
Min-Max	100	10	33,14	27,26	Standard Scaler	100	10	19,56	14,73
Min-Max	100	50	38,38	34,12	Standard Scaler	100	50	33,90	30,13

Based on Table 3, the RMSE and MAE values for each parameter with the RBF kernel were obtained, resulting in the best RMSE and MAE values when using Standard Scaler normalization, which had a lower average error compared to Min-Max normalization. The parameter combination [100, 10] yielded an RMSE value of 19.56 and an MAE value of 14.73.

For the best prediction results of Remaining Useful Life (RUL) for this turbofan engine, using the Radial Basis Function (RBF) kernel with parameters C [100] and epsilon [10], an RMSE of 19.56 and an MAE of 14.73 were obtained. A visualization of the prediction results and the actual values of the true RUL and the predictions in Figure 10 generated from the model training with normalization using Standard Scaler shows that the RBF kernel provides optimal results. This is due to the low error rate, as the lower the error, the more optimal the model is in predicting

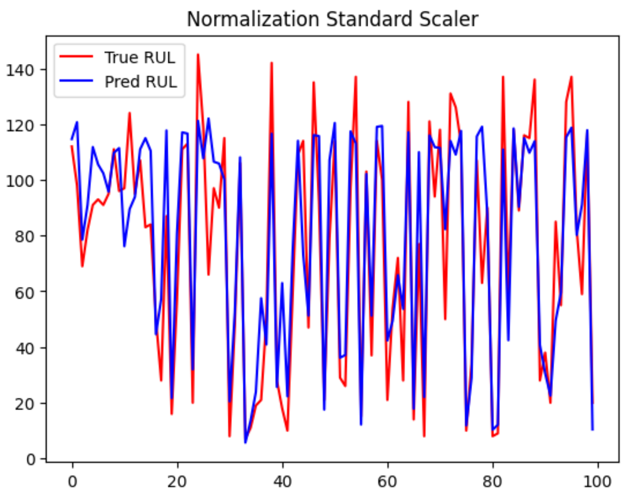

Figure 10. The best model from the Standard Scaler normalization method

CONCLUSION

Based on the research conducted, the use of the Support Vector Regression algorithm successfully predicted the Remaining Useful Life (RUL) of a turbofan engine using the C-MAPSS FD001 dataset. By utilizing the Radial Basis Function (RBF) kernel, optimal prediction results were achieved. The best model, obtained through Grid Search Optimization with C = 100 and epsilon = 10 and normalization using Standard Scaler, produced an RMSE of 19.56 and an MAE of 14.73. Furthermore, the values of C and epsilon can influence the model, potentially leading to overfitting. This experiment resulted in an optimal model visualization for predicting the machine's remaining useful life, indicating that the generated model was able to follow the pattern of RUL set in the RUL test.

REFERENCES

[1] T. Wang, "Trajectory Similarity Based Prediction for Remaining Useful Life Estimation," 2010.

[2] U. Thakkar, "Remaining Useful Life Prediction of a Turbofan Engine Using Deep Layer Recurrent Neural Networks," 2021.

[3] E. Priambodo, T. Prahasto, and A. Widodo, "Deteksi Dini Kerusakan Bearing Menggunakan Machine Learning Pendekatan Support Vector Regression (SVR)," 2023.

[4] O. Asif, S. A. L. I. Haider, S. R. Naqvi, J. F. W. Zaki, K. S. Kwak, and S. M. R. Islam, "A Deep Learning Model for Remaining Useful Life Prediction of Aircraft Turbofan Engine on C-MAPSS Dataset," *IEEE Access*, vol. 10, pp. 95425–95440, 2022, doi: 10.1109/ACCESS.2022.3203406.

[5] B. Maschler, *Continual Learning of Fault Prediction for Turbofan Engines using Deep Learning with Elastic Weight Consolidation*. IEEE, 2020.

[6] H. K. Wang, Y. Cheng, and K. Song, "Remaining useful life estimation of aircraft engines using a joint deep learning model based on tcnn and transformer," *Comput Intell Neurosci*, vol. 2021, 2021, doi: 10.1155/2021/5185938.

[7] D. Titian Wiranata, T. Prahasto, and A. Widodo, "Analisis Prognostik Terhadap Kerusakan Bantalan Pada Poros Kecepatan Tinggi Turbin Angin Menggunakan Machine Learning Dengan Pendekatan Support Vector Regression (SVR)," 2021.

[8] A. M. Siregar, S. Faisal, and B. Widiharto, "Model Prediksi Penderita Covid 19 Di Indonesia Menggunakan Metode Support Vector Regresion," 2022.

[9] Z. Rais, "Analisis Support Vector Regression (SVR) Dengan Kernel Radial Basis Function (RBF) Untuk Memprediksi Laju Inflasi Di Indonesia," *VARIANSI: Journal of Statistics and Its Application on Teaching and Research*, vol. 4, no. 1, pp. 30–38, 2022, doi: 10.35580/variansiunm13.

[10] R. P. Furi, M. S. Jondri, and D. Saepudin, "Prediksi Financial Time Series Menggunakan Independent Component Analysis dan Support Vector Regression Studi Kasus : IHSG dan JII," 2015.

[11] N. Asyraf *et al.*, "Peramalan Curah Hujan Di Kota Medan menggunakan Metode Support Vector Regression," 2022.

[12] D. K. Frederick, J. A. Decastro, and J. S. Litt, "User's Guide for the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS)," 2007. [Online]. Available: http://www.sti.nasa.gov

[13] E. Ramasso and A. Saxena, "Review and Analysis of Algorithmic Approaches Developed for Prognostics on CMAPSS Dataset," 2014.

[14] A. J. Smola, B. Sch¨olkopf, and S. Sch¨olkopf, "A tutorial on support vector regression," Kluwer Academic Publishers, 2004.

[15] A. Kazem, E. Sharifi, F. K. Hussain, M. Saberi, and O. K. Hussain, "Support vector regression with chaos-based firefly algorithm for stock market price forecasting," *Applied Soft Computing Journal*, vol. 13, no. 2, pp. 947–958, 2013, doi: 10.1016/j.asoc.2012.09.024.

[16] V. R. Prasetyo, B. Hartanto, and A. A. Mulyono, "Penentuan Pembimbing Tugas Akhir Mahasiswa Jurusan Teknik Informatika Universitas Surabaya Dengan Metode Dice Coefficient," *Teknika*, vol. 8, no. 1, pp. 44–51, 2019, doi: 10.34148/teknika.v8i1.147.

[17] M. Sholeh, D. Andayati, R. Yuliana Rachmawati, P. Studi Informatika, and F. Teknologi Informasi dan Bisnis, "Data Mining Model Klasifikasi Menggunakan Algoritma K-Nearest Neighbor Dengan Normalisasi Untuk Prediksi Penyakit Diabetes Data Mining Model Classification Using Algorithm K-Nearest Neighbor With Normalization For Diabetes Prediction," 2022.

[18] V. R. Prasetyo, M. Mercifia, A. Averina, L. Sunyoto, and B. Budiarjo, "Prediksi Rating Film Pada Website Imdb Menggunakan Metode Neural Network," *Network Engineering Research Operation*, vol. 7, no. 1, p. 1, 2022, doi: 10.21107/nero.v7i1.268.

[19] M. Raehanun, "Analisis Support Vector Machine (Svm) Dalam Prediksi," Yogyakarta, Jun. 2021.