ISSN 2278-3091
Volume 9, No.5, September - October 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse39952020.pdf
https://doi.org/10.30534/ijatcse/2020/39952020

Auto-feed Hyperparameter Support Vector Regression Prediction Algorithm in Handling Missing Values in Oil and Gas Dataset

Afnan Amirruddin[1], Izzatdin Abdul Aziz[2] & Mohd Hilmi Hasan[3]
[1,2,3]Centre for Research in Data Sciences (CeRDaS), UniversitiTeknologi PETRONAS, Malaysia,
afnan_18002965@utp.edu.my, izzatdin@utp.edu.my, mhilmi_hasan@utp.edu.my

ABSTRACT
There are many types of equipment involved in the oil and gas industry. However, they have their useful lives and will degrade over time. This issue prompts to be solved using predictive analytics to predict the Remaining Useful Life (RUL) of equipment. In the historical data, however, there are missing values due to broken equipment sensors probes and different time rate sensors. This can significantly affect the prediction results and making it less accurate due to missing value and become a challenging issue. Missing values in datasets is a synonymous problem in data mining which could lead to an incomplete dataset, making inaccurate predictions results in machine learning prediction processes. This problem inspires the idea to develop a prediction algorithm to predict the missing values in the dataset, where Support vector regression (SVR) has been proposed as a prediction method to predict missing values in several academic types of researches. SVR however is inferior in accuracy and thus this paper discusses the usage of an optimized SVR with Evolved Bat Algorithm (EBA) to handle the missing value accurately with high execution time. The paper also presents the topic of missing values in the dataset, as well as compares the performance of the optimized SVR with the original SVR in terms of accuracy and execution time while handling missing values in a large dataset. The novel optimization-based artificial intelligence algorithm proposed in this paper implies an improved way to overcome a real engineering challenge i.e. handling missing values for better RUL prediction, hence bringing great opportunities for the domain area.

Key words: Evolved Bat Algorithm, Machine Learning, Missing values, Oil and Gas, Remaining Useful Life, Support Vector Regression

1. INTRODUCTION

In the Oil and Gas industry, there are various pieces of equipment such as furnaces, storage tanks, heat exchanger, and many more that are used in extracting crude oil to produce refined oil. However, these types of equipment have their own useful lives and will degrade over time. Degradation occurs due to corrosion, deformation, fracture, and wear [1]. In a refinery, an unplanned shutdown due to equipment failure is a critical hit to the cost of operation in the refinery up to millions of dollars. The expense of unplanned shut down for a normal U.S. processing plant has been assessed at somewhere in the range of $340,000 and $1.7m per day [2].

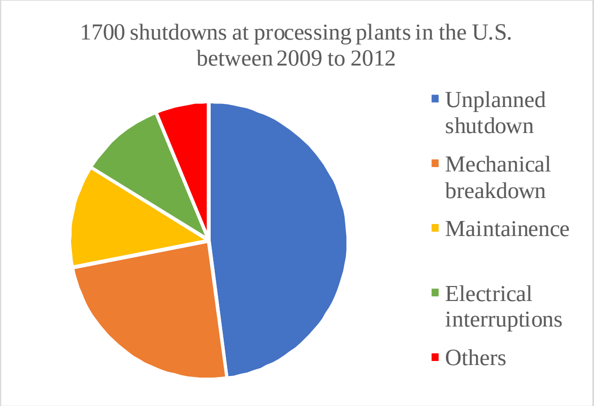


Figure 1: Shutdowns at processing plants in the U.S. between 2009 to 2012[2]

An investigation of the U.S. Department of Energy had found there were more than 1700 shutdowns at processing plants in the U.S. between 2009 and 2012. Of these, an expected 46 percent were because of mechanical breakdowns, 19 percent due to electrical interruptions, 23 percent on maintenance related issues, 12 percent for other causes such as a fire in the refinery, and a staggering amount of unplanned maintenance-related shut down by 92 percent [2]. This shows that the unplanned shutdown typically covers most of the shutdowns in a refinery. When shutdowns occur, the refinery could not process the crude oil and thus making the company that operates the company loses money due to the halt of the production of oil products. This also concludes that predicting the time of failure of equipment in refinery could help the company to have scheduled or planned shutdowns to reduce the number of unplanned shutdowns and thus saving more money.

This issue prompts the need for solutions using machine failure prediction techniques such as a machine-learning algorithm to predict the Remaining Useful Life (RUL) of the equipment. RUL is used to foresee the lifespan of parts to limit cataclysmic failure to occur in various parts of the refinery [1]. However, there is a risk in the process where the data is missing due to factors such as sensor malfunction. Thus, this creates missing values in a dataset.

Missing values in the dataset will spell inaccuracy in the prediction model [2] from problems such as data inconsistency and irrelevancy in the attributes of the data [3] and therefore, it is crucial to handle the missing values in the dataset to make the predictive analytics model more accurate or in the range of acceptable accuracy. According to [4], the problem of missing values can be resolved using machine learning methods and statistical methods. However, the selection of using a machine learning algorithm to handle the missing values in the dataset falls under the research domain compared to other methods, narrowing the scope of selection the method approach for the research. There are 3 types of missing values in data mining according to [5] which are Missing at Random (MAR), Missing Completely at Random (MCAR), and Not Missing at Random (NMAR).

There are several works of literature on Support Vector Regression (SVR) that have been proven can predict missing values. A paper by Honghai, F., et al. (2005) proposed the Support Vector Machine (SVM) Regression approach to handle missing values in a dataset [6]. A paper by Shi, W., et al. (2015) proposed the SVM variant prediction method to improve data quality in monitoring the power grid, which also covered the subject of handling missing values with the machine learning method [7]. In this paper, however, we are selecting a fast machine learning algorithm that is optimized with an optimization algorithm to handle the missing values.

There are three main objectives expected to be achieved by the end of the paper. Firstly, it is to study theoretically on the possible suitable machine-learning algorithm to predict missing values in terms of most accuracy value and execution time by conducting a literature review. Next is to develop the experiment with the chosen algorithm that has been optimized to predict missing values using a dataset from the local oil and gas company. The dataset would increment in value to simulate the usage of chosen algorithms from small to a larger dataset to observe the behavior of chosen algorithms with different sizes of the dataset. And the third objective is to evaluate the performance of algorithms and visualize the outcomes from the experiment.

This paper would utilize a real-time dataset by a local oil and gas company. The dataset is a time-series data that span in a year with 88 million rows and streamed from 17000 types of sensors in the equipment to the database and being processed by the oil and gas company. Subsequently, this research focuses on accomplishing low time execution for machine learning to handle missing value yet giving high accuracy prediction to be implied in the dataset.

2. LITERATURE REVIEW

2.1 Dataset Characteristics

Figure 2 represents the visual representation of the dataset in the seaborn heatmap format. The research will run the dataset by tags consisting of 'Tag i1', 'Tag i2', and 'Tag i3'. The dataset is historical time-series data that consisted of 2 million rows that were extracted to 200000 rows for the simulation purpose.

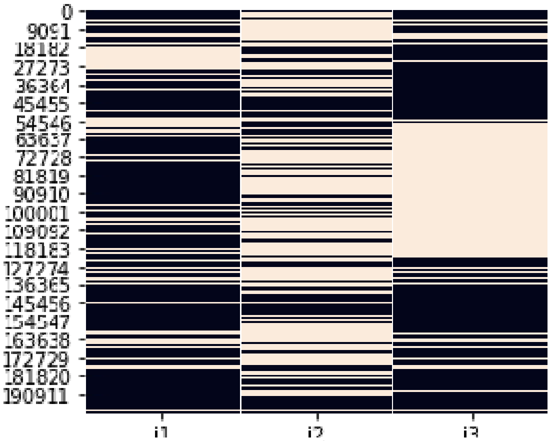


Figure 2: Seaborn heatmap of the dataset used

Also referring toFigure 2, we can observe that the dataset for the research containing 2 types of missing values which are MAR and MCAR where according to [5], MAR is defined as the probability of missing values on any attribute which is caused by the values of other attributes but not from its specific value while MCAR defined where the dataset has the highest frequency of randomness in the values that are missing. Besides, the probability of missing values on any other attribute does not depend on any value of the attribute. The other type of missing values which is NMAR is where the missing values depend upon the unavailable values.

2.2 Existing machine learning methods for predicting missing values

From literature specifically conducted a study on machine learning methods for predicting missing values, two machine learning methods have been used more frequently than the other methods to handle missing values. As studies and experiments conducted by [7,8] where both papers comparing SVR and ANN, these two algorithms have been chosen as two of the top contenders for the chosen methods to predict missing values out of many other machine learning methods available.

Some similarities of the algorithms are both of the algorithms are suitable for regression problems [9] which in the case of our dataset, regression capability is necessary and suitable. Besides, both of the algorithms also can be used for non-linear function [8]. Besides, both models can handle time-series problems [15].

The performance in terms of input sensitivity of the SVR algorithm performs better than the ANN model, thus making it more robust [15]. The performance of SVR is more accurate with fewer input variables and less accurate with more input variables and vice versa for ANN [8]. Some of the advantages of SVR is the algorithm is more robust and more sensitive to input variables compared to ANN. However, ANN is a better predictor universally compared to SVR [8].

In terms of speed, SVR takes a shorter time to execute. It would be better for larger datasets compared to ANN [9]. As from an experiment by [9], the difference for SVR and ANN in the time taken for execution is around 18ms difference for a single sample. However, due to the case study having a large dataset, the time execution is the focus in choosing the algorithm to be optimized by the optimization algorithm. Also, in the research scope of the whole RUL prediction research, the dataset would need to be refreshed every 10 minutes and the time window of 10 minutes to execute the process of data pre-processing, missing value prediction and predict the expected RUL. Thus, choosing a faster algorithm, which in this case is SVR in the work process of filling the missing values in the dataset for the research domain is a suitable choice.

2.3 SVR

SVR, together with Support Vector Classification (SVC) are the categories of Support Vector Machine (SVM). SVM is a learning framework utilizing a high dimensional feature space and a prediction algorithm that works based on mathematics for enhancing a mathematical function regarding a collected dataset. The basic contemplations behind the SVR algorithm can be explained without technical equations of the algorithm [10]. To comprehend the fundamental of SVR, there are four basics fundamental in SVR that must be grasped according to [11]:

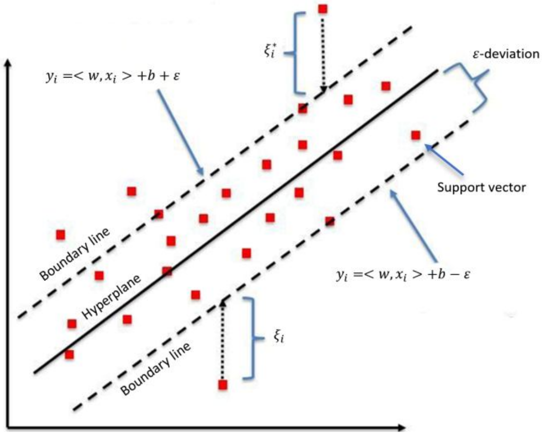


Figure 3: SVR Algorithm[11]

where SVR consists of four main components which are hyperplane, kernel function, support vectors, and boundary lines [11]. The hyperplane is a guiding line that helps to predict the target or continuous value. The hyperplane lines with the maximum number of points will be the best fit line [11]. Boundary lines are the lines that keep the margin of the predictions. It also represents the distance of the lines in the equation and is known as epsilon, ε[11]. Next, a kernel function is a function in the algorithm to map higher dimensional data from lower dimensional data [11]. There are 4 main kernels for the algorithm as listed below:

1. Linear kernel:

$$k(x_i, x_j) = x_i^T x_j \tag{1}$$

2. Polynomial kernel:

$$k(x_i, x_j) = (\gamma x_i^T x_j) \tag{2}$$

3. Gaussian or RBF (radial bias function) kernel:

$$k(x_i, x_j) = \exp\left(-\gamma \|x_i - x_j\|^2\right), \gamma > 0 \tag{3}$$

4. Sigmoid kernel:

$$k(x_i, x_j) = \tanh\,(\gamma x_i^T x_j + r) \tag{4}$$

Lastly is support vectors, which are the data points that are nearest to the boundary lines [11]. SVR uses hyperparameters to calculate Radial Basis Function (RBF) kernel, linear kernel, and polynomial kernel and makes a prediction and chooses suitable data for the dataset. However, SVR does not perform well with a larger dataset and multivariate analysis but it is more sensitive to individual input variables [12].

To integrate to the research, we are applying the SVR using training dataset $\{(x_1, y_1), (x_2, y_2), \ldots\ldots, (x_n, y_n)\}$ and where $x_i \in R^n$is the input or in this research is the existing values, and $y_i \in R^1$ is the output value for the predicted result [13]. Therefore, the equation of estimation function $f(x)$ can be written as follows:

$$f(x) = \omega \cdot \psi(x) + b \tag{5}$$

SVR algorithm finds to predict $f(x)$, $\omega \in R^n$ and $b \in R$[13][6] by reducing the regularized risk function where ω is a weight vector, and b is the bias term [13].

The regression problem in SVR can be converted to the optimization problem using the ε-insensitive lost function where ε is the tube radius or tolerance margin, which alludes to the data inside the tube that ought to be disregarded during the regression process. The feature vector that is lying on the tube boundary becomes the support vector. Applying a Lagrange multiplier method to the equation, the following equation could be achieved:

$$min\colon \frac{\|\omega\|^2}{2} + C \sum_{i=1}^{n} (\xi_i - \xi_i^*) \tag{6}$$

$$\begin{aligned} max\colon \overline{\omega}(\alpha, \alpha^*)_{\varpi,b,\xi,\xi^*} &= \frac{1}{2} \sum_{i,j=1}^{n} (\alpha_i - \alpha_i^*)(a_j - a_j^*) \\ &< \psi(x_i), \psi(x_j) > \\ - \sum_{i=1}^{n} (\alpha_i + \alpha_i^*)\varepsilon &+ \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) y_i, \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) \\ &= 0; 0 \leq \alpha_i, \alpha_i^* \leq C \end{aligned} \tag{7}$$

where C will be the determinant of the estimation errors [13]. Eventually, the equation is modified by the substitution of the kernel function $k(x_i, x_j)$ to replace $< \psi(x_i), \psi(x_j) >$ and the final equation can be outlined:

$$\begin{aligned} max\colon \overline{\omega}(\alpha, \alpha^*)_{\varpi,b,\xi,\xi^*} & \\ &= \frac{1}{2} \sum_{i,j=1}^{n} (\alpha_i - \alpha_i^*)(a_j \\ &- a_j^*) k(x_i, x_j) - \sum_{i=1}^{n} (\alpha_i + \alpha_i^*)\varepsilon \\ &+ \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) y_i, \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) \\ &= 0; 0 \leq \alpha_i, \alpha_i^* \leq C \end{aligned} \tag{8}$$

The comprehension basic of the statistical learning theory is that to procure a little risk where the foundation of model complexity and training error is required to be control of [6]. The kernel function is one of the main variables of SVR. SVR performance is to a great extent, rely on the kernel selection and the parameters. The results of the calculation will be predicted using the RBF kernel as shown previously in (3). RBF is used as the kernel in this methodology because it requires only one parameter and it has a wide scope of application. RBF kernel also can universally approximate any distribution in feature space [13].

2.4 Optimization Algorithm

An optimization algorithm is vastly available in many situations and applications from cost optimization, consumption of energy, performance, and efficiency of any scope. The ideal utilization of the accessible resource of any kind requires a change in perspective in logical reasoning. This is due to the majority of applications in the real-world situation that have undeniably progressively confounded variables and parameters to influence how the system acts [14].

Search optimization algorithms are the instruments and methods of accomplishing the optimality of any related problem. Optimal solutions are not practical in real-life applications however, due to the solutions being not robust enough, while suboptimal solutions are often being chosen for a good robust solution in such problems [14].

Thus, the idea of developing an algorithm with a search optimization property could handle the problem which the original SVR algorithm is lacking, in terms of the average accuracy of the prediction results. Some of the algorithms that have been chosen to be compared for the selection of the most suitable algorithm to combat the disadvantage of the original SVR algorithm. The optimization algorithm has been chosen after referring to some of the academic papers related to optimization problems such as [15], [16], [17], [18], [19],[20] and [21].

2.5 Evolved Bat Algorithm

A novel algorithm that optimized based on the Bat Algorithm (BA) that has been suggested by [15] named Evolved Bat Algorithm (EBA) is an algorithm that has been tuned for optimizing numerical problems. By re-evaluating the bats and having options on the general qualities of the entire species of the bats, [15] reclassifies the responsible operations to the behavior of the bats. EBA is a novel technique within the swarm intelligence to challenge the optimization in numerical problems [15].

For EBA, bat movement and the random walk process has been optimized from the original BA. The fixed value of the sound speed in the air is 340 meters per second (m/s) and the distance between the target and the source of the sound wave, which the wave is bounded is defined using the following equation:

$$D = \frac{v \cdot \Delta T}{2} \qquad (9)$$

which D is for the distance, v denotes the speed of sound, and ΔT representing the difference of time between the wave of sound and the echo receiver. According to [15], the estimated value of v is the speed of the sound in the air. Moreover, [15] substitute the metric unit of v m/s to kilometer per second (km/s). Therefore, (9) can be updated to (10):

$$D = 170 \cdot \Delta T(\frac{m}{s}) \rightarrow 0.17 \cdot \Delta T(\frac{km}{s}) \qquad (10)$$

In the experiment that has been conducted by [15], the value of ΔT is a random number, $n \in [-1,1]$. The selection of random number which involves with negative number is due to the coordinate movement and the direction of transmission being the sound wave traveling from the opposite to the coordinate axis. The bat movement can be formulated as follows:

$$x_i^t = x_i^{t-1} + D \qquad (11)$$

and the location will be updated as shown in (12) as the bat engaged in the random walk process:

$$x_i^{t_R} = \beta \cdot (x_{Best} - x_i^t) \qquad (12)$$

which β denotes a random number where $\beta \in [0,1]$ and $x_i^{t_R}$ represents the updated bat location after the random walk process.

As mentioned in [15], EBA is suitable for optimizing SVR due to the ability of EBA to find the most appropriate value to fill in the hyperparameter value. The SVR will be optimized in the kernel initialization process. EBA will use its ability to generate a random number and if the number is larger than the emission rate, it would move the bat with the random walk process. The bat then will be evaluated and updated [15]. The value then, considering that it is the best value for the kernel, will be updated to the kernel. The kernel then will do the prediction for missing values. Because of the hyperparameter tuning, the results will be better than the original SVR without optimization. The optimization of SVR with EBA is appropriately named as Auto-feed Hyperparameter SVR (AHSVR).

3. METHODOLOGY

The objectives of these simulations are as follows:

- To measure the prediction accuracy of both SVR and AHSVR to predict the missing values on a time-series dataset.
- To measure the time execution of both SVR and ANN in handling large time-series data with differentpercentages of missing values.

The expected outcomes from the simulations are the comparison between SVR and the optimized AHSVR prediction algorithm in terms of accuracy and speedup, where the discussions over simulations results are discussed and explained in Section 4.

3.1 AHSVR

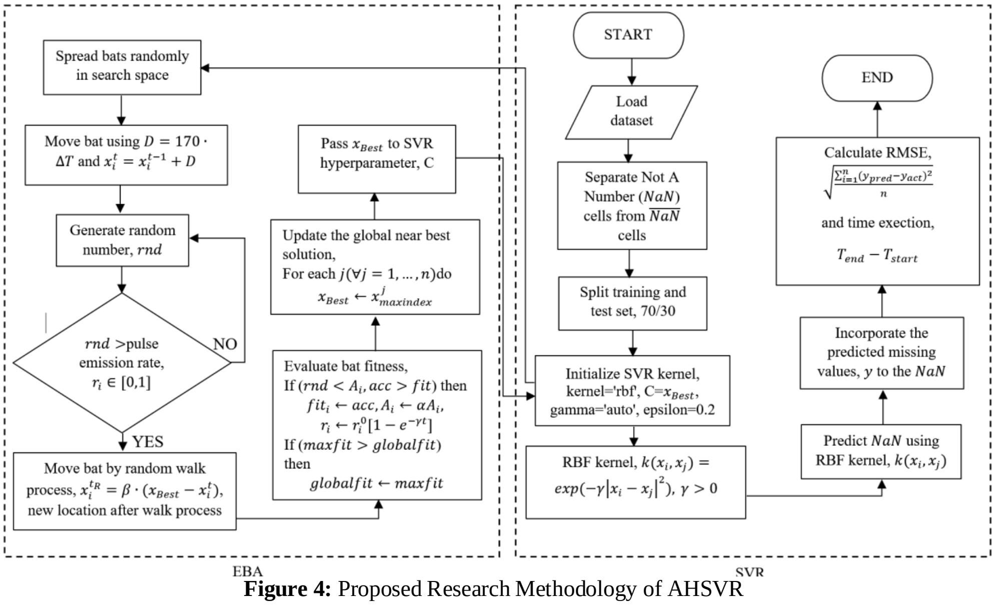

Spread bats randomly in search space
Move bat using $D = 170 \cdot \Delta T$ and $x_i^t = x_i^{t-1} + D$
Generate random number, rnd
rnd >pulse emission rate, $r_i \in [0,1]$
NO
YES
Move bat by random walk process, $x_i^{t_R} = \beta \cdot (x_{Best} - x_i^t)$, new location after walk process
Evaluate bat fitness, If ($rnd < A_i, acc > fit$) then $fit_i \leftarrow acc, A_i \leftarrow \alpha A_i,$ $r_i \leftarrow r_i^0[1 - e^{-\gamma t}]$ If ($maxfit > globalfit$) then $globalfit \leftarrow maxfit$
Update the global near best solution, For each $j(\forall j = 1, \dots, n)$do $x_{Best} \leftarrow x_{maxindex}^j$
Pass x_{Best} to SVR hyperparameter, C
EBA
START
Load dataset
Separate Not A Number (NaN) cells from $\overline{NaN}$ cells
Split training and test set, 70/30
Initialize SVR kernel, kernel='rbf', C=x_{Best}, gamma='auto', epsilon=0.2
RBF kernel, $k(x_i, x_j) = exp(-\gamma|x_i - x_j|^2), \gamma > 0$
Predict NaN using RBF kernel, $k(x_i, x_j)$
Incorporate the predicted missing values, y to the NaN
Calculate RMSE, $\sqrt{\frac{\sum_{i=1}^{n}(y_{pred}-y_{act})^2}{n}}$ and time exection, $T_{end} - T_{start}$
END
SVR

Figure 4: Proposed Research Methodology of AHSVR

For the process of loading the dataset, the execution is looped from 20000 rows to 200000. This step is to simulate the condition where we could evaluate the performance of the algorithm to different sizes of the dataset. The accuracy, however, depending on the ratio of missing values for each tag. Therefore, the execution also running by each tag in a loop. In the next process, the separation between NaN and $\overline{NaN}$ is necessary for the algorithm to identify the difference between our target values, y and input values, x. Within the separation, we could use input values to train the model to predict the y.

For the SVR algorithm initialization, the RBF kernel is a suitable candidate as it is often used as the kernel in SVR because it requires one parameter only and it has board scope of application for prediction. RBF also works best in feature space as it can universally approximate every type of distribution [13]. For the initialization process, RBF kernel required values for hyperparameter C, gamma, γ and epsilon, e. γ has been set to auto configuration and e has been set to 0.1 by default [22]. For parameter C however, the EBA algorithm is used to find the optimal value of the parameter.

The first step in EBA is to populate the search space by spreading the bat to it. Few variables were used for the initialization process of EBA that is different from the original BA as listed inTable 1:

Table 1: Initialization of unique variables

User-defined variables	Value
Sound speed, v	340 m/s
Time difference, ΔT	$\Delta T \in [-1{,}1]$
Random number, β	$\beta \in [0{,}1]$
Loudness, A	$A \in [1{,}2]$
Fixed pulse emission rate, r	0.5

After initialization, we can move the bat throughout the search space. The movement of the bats is determined with previously mentioned (10) where the equation determined the value of distance, D which calculated and replaced to (13):

$$x_i^t = x_i^{t-1} + \left(17 \cdot \Delta T \left(\frac{km}{sec}\right)\right) \tag{13}$$

where it will calculate the movement of the bat. The algorithm then would generate a random number, β. For a typical bat algorithm, the algorithm would compare in a condition where if β is greater than the pulse emission rate r_i, where the i represents the i^{th}. If the condition is fulfilled, the random number will be selected. This process is repeated for each β generated if the condition does not fulfill. However, in EBA, the value for pulse emission rate, r is fixed to 0.5 [15] and the same comparison will be executed. This process is repeated for each β generated if the condition does not fulfil. After the selection, the chosen β will be inserted into the random walk process calculation which is outlined previously in (12)where the new location of the bat is represented by $x_i^{t_R}$. For the next step, the bat needs to be evaluated in terms of fitness [15].

The model is then needed to be validated by predicting the NaN values, y to the incomplete dataset. The model, however, needs to be measured in terms of accuracy and its execution time. Root Mean Squared Error (RMSE), which is typically used regression metrics suggested by [22] for performance accuracy evaluation and speedup are the preferred methods to calculate the accuracy and time execution of the model.

RMSE is a function that calculates the root of the mean square error, which is a metric of risk related to the estimated value of the loss or squared error which is quadratic [22]. The calculation is as shown in (14):

$$RMSE(y, \hat{y}) = \sqrt{\frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} (y_i - \hat{y}_i)^2} \tag{14}$$

where $\hat{y_i}$ is the target value for ith sample and y_i is the relating true value.

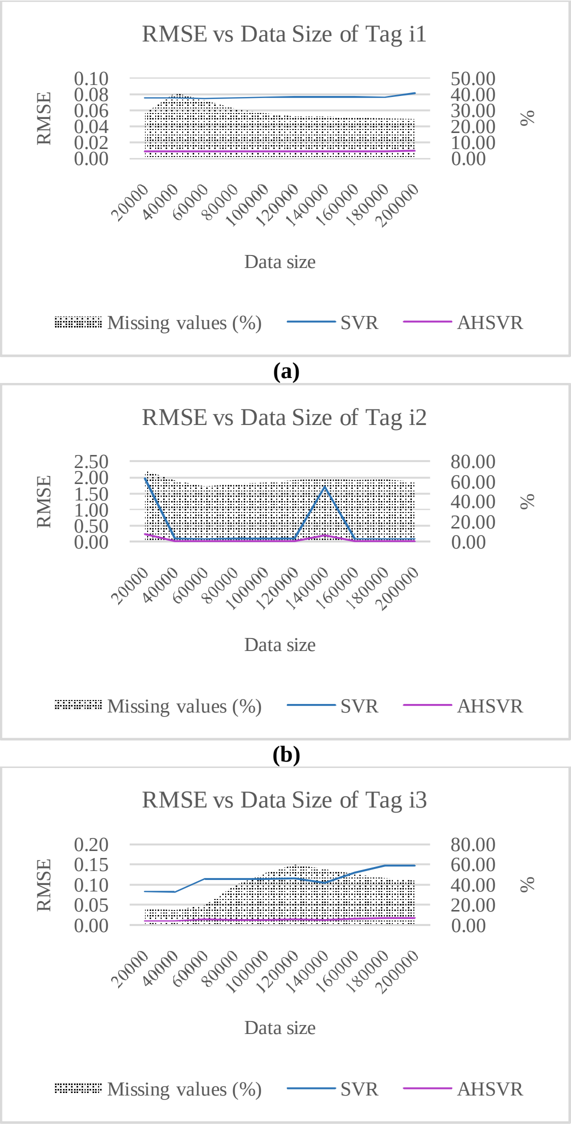
The simulation has been conducted to evaluate the performance of SVR and ANN algorithms in terms of speed efficiency using the speedup metrics. The speedup was calculated using this equation:

$$S_{latency} = \frac{T_{SVR}}{T_{AHSVR}} \qquad (16)$$

where T_{SVR} represents the time execution of SVR and T_{AHSVR} represents the time execution of AHSVR.

4. RESULTS AND DISCUSSION

The simulations have been conducted using the metrics and the results have been recorded and outlined:

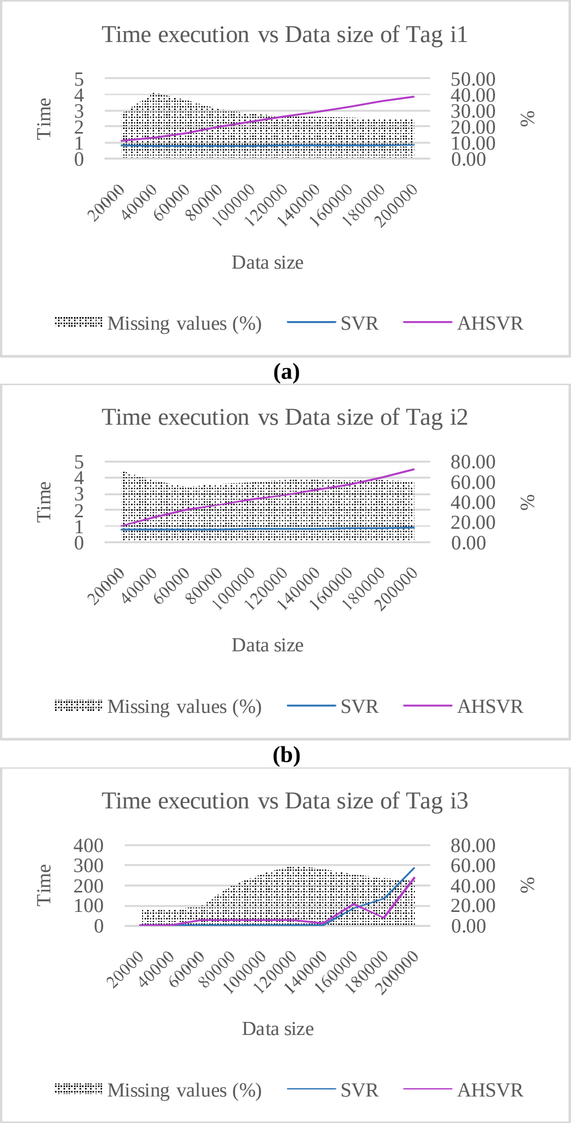


(a)

(b)

(c)

Figure 5: Performance accuracy evaluation of SVR and AHSVR algorithm using RMSE



(a)

(b)

(c)

Figure 6: Performance time execution of SVR and AHSVR algorithm

Table 2: Speedup of AHSVR over SVR

Rows	Tag i1	Tag i2	Tag i3
	Speedup	Speedup	Speedup
20000	1.38688538	1.29277201	1.58273745
40000	1.67416931	1.98835746	2.22489649
60000	2.05008775	2.57967569	20.23213645
80000	2.50522795	2.98608520	24.10304692
100000	2.91206144	3.36824760	19.21894681
120000	3.25009137	3.68750162	26.38588155
140000	3.61109663	4.09103731	4.06181702
160000	3.94387638	4.31689614	1.24603446
180000	4.38132723	4.69074348	0.28719909
200000	4.53389388	5.05774225	0.83188554
Total	30.24871733	34.05905876	100.17458178
Average	3.02487173	3.40590588	10.01745818
Total average	≈5.48x speedup over AHSVR		

Figure 5 represents the RMSE performance evaluation metric that has been executed to observe the improvement of the optimized SVR which is AHSVR, compared to the performance accuracy of SVR. We can observe that in Figure 5 (a) and (c) where the performance differences of the two algorithms are vast, where AHSVR came with a significant difference of around 80% better compared to the SVR. The results also illustrated that AHSVR made significant improvement in terms of accuracy compared to SVR, validating that the optimization of the SVR algorithm in the hyperparameter can increase the accuracy of the algorithm significantly albeit some hiccups over the presence of a significant amount of missing values, which are applicable for both SVR and AHSVR. This is credited to the capability of EBA to find the optimal number [15] for the hyperparameter value of SVR to optimize the performance of AHSVR in every missing value situation. In Figure 5 (b), it demonstrated for both algorithms that a high number of MAR values and MCAR in our dataset resulting our independent variables significantly lesser to find our target value, making the accuracy low.

Referring to Figure 6, we can observe that the higher the data size, resulting in a higher time execution for the algorithm. Besides, AHSVR requires more execution time for initialization and bat processes, also with repeating for every execution of different data size making the algorithm took a longer compilation and execution. In Figure 6 (c), we can observe that the execution time of the algorithms is inconsistent compared to Figure 6 (a) and (b). It contained a significant amount of missing values in the dataset or MCAR which could impact the performance of the prediction algorithm in terms of time execution. The cumulative missing values from 120000th row making both SVR and AHSVR require more processing power and thus, making the performance increase after the cumulative 120000th row as the algorithm needs to train a more complete dataset for better performance in time and accuracy.

We also evaluated the differences between SVR and AHSVR in terms of performance execution time with the speedup. With speedup, the ratio did not depend on the factor such as hardware performance and background tasks making the ratio free from unwanted variables that could lead to the inconsistent reading of the time execution. The observation can be made in Table 2 where the performance of SVR is approximately 5.48 times over AHSVR. This validates that the optimization of SVR does not jeopardize the time execution of the algorithm.

To conclude, a higher percentage of missing values in a cumulative dataset cell significantly exert influence on the prediction accuracy in missing value prediction accuracy while in terms of time execution, the significant missing values in the dataset affect the processing power for the hardware, making the predictive algorithm took more time for execution compared to a more consistent missing value dataset.

5. CONCLUSIONS

The problems concerning SVR in prediction to handle missing value in time-series data have been studied. The comparison study of the existing machine learning algorithms that have been conducted by various scholars also has been collected and documented in this research study.

Firstly, the objective of the research was to investigate the existing machine learning algorithms that can be used to predict missing values by using the time-series dataset and the study was outlined in the literature review. The second objective was to develop the optimized algorithm for simulation using the local oil and gas dataset. And the final objective was to evaluate the performance of the proposed optimized method in terms of execution time and accuracy in prediction using speedup and performance evaluation metrics.

A simulation study has been conducted to represent the differences between SVR and AHSVR in this research in terms of predicting missing values in time-series data. The results had highlighted that AHSVR performed better compared to SVR in terms of performance accuracy, with acceptable execution time. Both algorithm also can be utilized to do prediction on time-series data with the large dataset by getting a feasible result in regards of time efficiency and accuracy has been demonstrated, the correlation between the data size and performance of both algorithms also has been outlined and the accuracy and time efficiency of both algorithms has been shown.

From this research, we concluded that the method of handling missing values could be accomplish using the machine learning method. Also, we can optimize a fast executing algorithm to enhance the accuracy of the algorithm to suit our usage where there is a short time window to do the imputation to a dataset, thus producing a novel machine learning algorithm for various application in data science as well as industries such as engineering applications.

ACKNOWLEDGMENT

This work is one of the ongoing researches supported by the Centre for Research in Data Sciences (CeRDaS), UniversitiTeknologi PETRONAS, Malaysia

REFERENCES

1. C. Okoh, R. Roy, and L. Redding, “Overview of Remaining Useful Life Prediction Techniques in Through-life Engineering Services,” Procedia CIRP, vol. 16, pp. 158–163, 2014, doi: 10.1016/j.procir.2014.02.006.
2. Elsevier’s R&D Solutions, “Challenges in Achieving Operational Excellence in Refining & Petrochemicals,” R&D Solut. Oil Gas, p. 7, 2016, Accessed: Sep. 24, 2019. [Online]. Available:

https://www.elsevier.com/__data/assets/pdf_file/0009/230868/RDS_OG_RP_WP_OPEX-in-Refining-Petrochemical_DIGITAL.pdf.

3. J. S. Gil, A. J. P. Delima, and R. N. Vilchez, "Predicting students' dropout indicators in public school using data mining approaches," Int. J. Adv. Trends Comput. Sci. Eng., vol. 9, no. 1, pp. 774–778, 2020, doi: 10.30534/ijatcse/2020/110912020.
4. A. Shah, "Machine Learning vs Statistics," KDnuggets, 2016. https://www.kdnuggets.com/2016/11/machine-learning-vs-statistics.html (accessed Feb. 18, 2019).
5. L. Peng and L. Lei, "A Review of Missing Data Treatment Methods," An Int. J. Intell. Inf. Manag. Syst. Technol., vol. 1, no. 3, p. 17, 2005, Accessed: Feb. 07, 2019. [Online]. Available: https://pdfs.semanticscholar.org/e4f8/1aa5b67132ccf875cfb61946892024996413.pdf.
6. F. Honghai, C. Guoshun, Y. Cheng, Y. Bingru, and C. Yumei, "A SVM Regression Based Approach to Filling in Missing Values," in 9th International Conference, KES 2005 Melbourne, Australia, September 14-16, 2005 Proceedings, Part III, 2005, pp. 581–587, Accessed: Feb. 26, 2019. [Online]. Available: https://link.springer.com/content/pdf/10.1007%2F11553939.pdf.
7. W. Shi et al., "Improving power grid monitoring data quality: An efficient machine learning framework for missing data prediction," Proc. - 2015 IEEE 17th Int. Conf. High Perform. Comput. Commun. 2015 IEEE 7th Int. Symp. Cybersp. Saf. Secur. 2015 IEEE 12th Int. Conf. Embed. Softw. Syst. H, pp. 417–422, 2015, doi: 10.1109/HPCC-CSS-ICESS.2015.16.
8. A. Shirzad, M. Tabesh, and R. Farmani, "A Comparison between Performance of Support Vector Regression and Artificial Neural Network in Prediction of Pipe Burst Rate in Water Distribution Networks," KSCE J. Civ. Eng., vol. 18, no. 4, pp. 941–948, 2014, doi: 10.1007/s12205-014-0537-8.
9. A. Ameri, E. N. Kamavuako, E. J. Scheme, K. B. Englehart, and P. A. Parker, "Support vector regression for improved real-time, simultaneous myoelectric control," IEEE Trans. Neural Syst. Rehabil. Eng., vol. 22, no. 6, pp. 1198–1209, 2014, doi: 10.1109/TNSRE.2014.2323576.
10. W. S. Noble, "What is a support vector machine?," Nat. Biotechnol., vol. 24, no. 12, pp. 1565–1567, 2006, doi: 10.1038/nbt1206-1565.
11. I. Bhattacharyya, "Support Vector Regression Or SVR," Medium.com, 2018. https://medium.com/coinmonks/support-vector-regression-or-svr-8eb3acf6d0ff (accessed Apr. 29, 2019).
12. N. Kuruwitaarachchi, M. K. M. Peiris, C. N. Madawala, K. M. A. R. Perera, and V. U. N. Perera, "Design and Development of an Algorithm to Predict Fluctuations of Currency Rates," in 11th International Conference on Software, Knowledge, Information Management & Applications, At Colombo, 2017, no. December 2017, p. 7.
13. U. K. Das et al., "SVR-Based Model to Forecast PV Power Generation under Different Weather Conditions," World Acad. Sci. Eng. Technol. Int. J. Comput. Inf. Eng., vol. 12, no. 2, p. 17, 2018, doi: 10.3390/en10070876.
14. X. Yang, Optimization and Metaheuristic Algorithms in Engineering, in: Metaheursitics in Water, Geotechnical and Transport Engineering. Elsevier, Inc, 2013.
15. P. W. Tsai, J. S. Pan, B. Y. Liao, M. J. Tsai, and V. Istanda, "Bat algorithm inspired algorithm for solving numerical optimization problems," Appl. Mech. Mater., vol. 148–149, no. December, pp. 134–137, 2012, doi: 10.4028/www.scientific.net/AMM.148-149.134.
16. X. S. Yang, "Bat algorithm: Literature review and applications," Int. J. Bio-Inspired Comput., vol. 5, no. 3, pp. 141–149, 2013, doi: 10.1504/IJBIC.2013.055093.
17. V. Selvi and R. Umarani, "Comparative Analysis of Ant Colony and Particle Swarm Optimization Techniques," 2010. Accessed: Mar. 12, 2019. [Online]. Available: https://s3.amazonaws.com/academia.edu.documents/34460120/pxc3871286.pdf?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1552323586&Signature=9pIwvgkoHLf5s%2Fsg%2BHRXHHZvVRc%3D&response-content-disposition=inline%3B filename%3DComparative_Analysis_of_Ant_Colon.
18. M. Jiang, S. Jiang, L. Zhu, Y. Wang, W. Huang, and H. Zhang, "Study on Parameter Optimization for Support Vector Regression in Solving the Inverse ECG Problem," Comput. Math. Methods Med., vol. 2013, no. July, p. 9, 2013, doi: 10.1155/2013/158056.
19. M. Tabassum and K. Mathew, "A Genetic Algorithm Analysis towards Optimization solutions," Int. J. Digit. Inf. Wirel. Commun., vol. 4, no. 1, pp. 124–142, 2015, doi: 10.17781/p001091.
20. T. Kerdphol, K. Fuji, Y. Mitani, M. Watanabe, and Y. Qudaih, "Optimization of a battery energy storage system using particle swarm optimization for stand-alone microgrids," Int. J. Electr. Power Energy Syst., vol. 81, pp. 32–39, 2016, doi: 10.1016/j.ijepes.2016.02.006.
21. F. H. Mausor, J. Jaafar, and S. Mohdtaib, "Fuzzy C means imputation of missing values with ant colony optimization," Int. J. Adv. Trends Comput. Sci. Eng., vol. 9, no. 1 Special Issue 3, pp. 145–149, 2020, doi: 10.30534/ijatcse/2020/2191.32020.
22. F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011, Accessed: Aug. 21, 2019. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html.