

Predicting machine failure using recurrent neural network-gated recurrent unit (RNN-GRU) through time series data

Zainuddin Z., P. Akhir E. A., Hasan M. H.

Department of Computer & Information Science, Universiti Teknologi Petronas, Tronoh Perak, Malaysia

Article Info

Article history:

Received Oct 25, 2020

Revised Jan 17, 2021

Accepted Feb 15, 2021

Keywords:

Gated recurrent unit
Machine failure
Machine learning
Prediction
Time series data

ABSTRACT

Time series data often involves big size environment that lead to high dimensionality problem. Many industries are generating time series data that continuously update each second. The arising of machine learning may help in managing the data. It can forecast future instance while handling large data issues. Forecasting is related to predicting task of an upcoming event to avoid any circumstances happen in current environment. It helps those sectors such as production to foresee the state of machine in line with saving the cost from sudden breakdown as unplanned machine failure can disrupt the operation and loss up to millions. Thus, this paper offers a deep learning algorithm named recurrent neural network-gated recurrent unit (RNN-GRU) to forecast the state of machines producing the time series data in an oil and gas sector. RNN-GRU is an affiliation of recurrent neural network (RNN) that can control consecutive data due to the existence of update and reset gates. The gates decided on the necessary information to be kept in the memory. RNN-GRU is a simpler structure of long short-term memory (RNN-LSTM) with 87% of accuracy on prediction.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Zainuddin Z
Computer & Information Science Department
Universiti Teknologi Petronas
Bandar Seri Iskandar, 32610 Tronoh, Perak
Email: zahirah_18003491@utp.edu.my

1. INTRODUCTION

Oil and gas industry deals with lots of activities such as manufacturing the oil and gas for sale while providing some sorts of services in refining the oil and gas and transport them to a required petrol station [1]. All the activities generate time series data from sensors, corporate document archive and internet. Time series data is a successive of data points observed over a chronological of time [2]. The data is updated each second and revolves around big data size issue and complexity of attributes [3]. This research adopts a set of time series data apprehended by an oil and gas corporation generated from several kinds of sensors. The data is given in massive amount of 55 GB kept in CSV format. There are three types of columns consisting tag (name of machines), time and value with an interval of one year. The information is increasing over time lead to big data size problem. They in need of new algorithms and procedures to handle them for getting a new result [4]. Hence, machine learning (ML) algorithm is proposed to manage the problem.

ML is an aid to computer by modeling past experiences for forecasting the future considering as a major topic in artificial intelligence (AI) [5]. It has been an incredible discover in identifying the connection among the information, processing big data size and able to achieve the same performance as machine operation [6]. The algorithm is valuable when it receives more data as it can grasp the pattern from them to predict on new outcome. Prediction is defined as an activity of grapping the acquired knowledge to process

them and result in the unknown information [7]. It is implemented in variety of fields for example financial, sales and many more. The term helps an organization from bearing a responsible for the loss incurred in case of machine fault.

A fault can be an unusuality showed by the machine or gears such as any component undergoes sudden breakdown or shutdown [8]. A fault occur to the machine is the root cause of machine failure. Immerman [9] stated that 82% of firms suffer losses up to \$260.000 per hour when machine faced with sudden fail for the past three years. Stratus technologies calculated almost \$20 Billion or 5% from the total production lost during unexpected failure [10]. Thus, predicting machine failure can help monitoring the machine condition and sparing the expenses as they will be examines over time [8]. Thus, this paper is focusing on predicting machine failure by using time series data induced from the machine itself to achieve high availability in production process and aim for zero unexpected failure. The prediction is constructed by using ML algorithm in regard to my domain in information technology (IT).

ML can run time series data to locate the pattern for forecasting purpose [11]. The author proved by applying random forest (RF) to prognosticate sales. The sales distribution is analysed to conceptualize the pattern of sales upon years. Once RF is attached to the data, the error estimation is tabulated. Mean absolute error (MAE) is referred as the error calculation in (1).

$$\text{Error} = \frac{\text{MAE}}{\text{Mean(sales)}} * 100\% \quad (1)$$

The quotient of training is recorded to 3.9% while the validation hits 11.6%. Then, RF is generalized to prevent bias occasion. The generalization helps in getting more exact result from the pattern showed by the data even though the presence of noise is detected [11].

Another research from [12] that utilized deep learning (DL) to perform on time series data in solar photovoltaic (PV). The data is modified according to previous structure and forecasting scope as the scope will react to each problem. Then, the data is divided into three categories called training, testing and validation. DL has its own parameters that affect the sequel thus grid search is imposed by using the training and validation sets. The search will find the ideal values for the parameters as the solution and being compared with the validation set. The most optimal one will be used for foreseeing task. DL displayed the lowest root mean squared error (RMSE), 148.98 and 114.76 for MAE compared to the other algorithms. Therefore, ML works well with time series data for prediction.

Rebala *et al.* [13] categorized three types of ML algorithms named supervised, unsupervised and reinforcement. Supervised is the relation between input and output through a function produced by the algorithm. In other hand, unsupervised is dealing with unmarked data which required the model to self-learn by its own to attain information. Reinforcement is the response of algorithm with reference to benefit and penalization. Praveena and Jaiganesh [14] supports the statement by illustrating machine learning algorithms into three classes as per Figure 1.

Figure 1 shows that supervised is a task driven or a series of activity that being fulfilled to satisfy the objectives. The task driven can be classified into regression or classification. Regression is working on continuous data while classification implies at a fixed esteem point [15]. The second category is known as unsupervised or data driven group. Data driven is an expression for an activity influenced by data and did not impacted by instinct or individual observation. The most common example is clustering. Clustering allows one to search for the group with similar characteristics in a partition considering them belongs to unsupervised learning style [16]. The last class is reinforcement which permits the algorithm to react towards the world's perception depending on policy learned. The surrounding will be affected by the reaction thus encompassing responses aided on the algorithm.

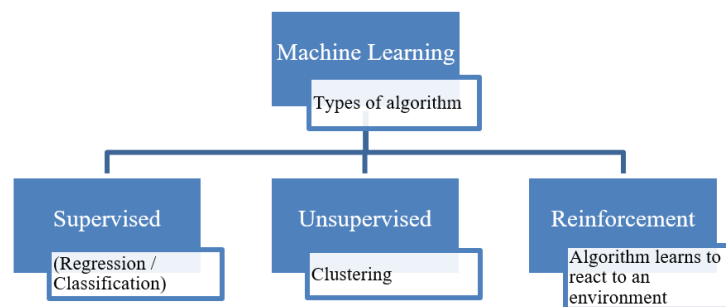
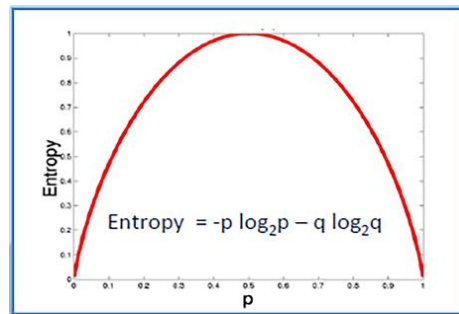


Figure 1. Types of machine learning algorithm

Hence, this research is contemplating to regression task under supervised division. This is because regression task refers to the model changing in forecasting numerical value [17] that fits the focal point of the research such prediction. In other words, this research will predict the remaining life of machine from an oil and gas company by using the numerical data produced by them. Iqbal and Yan [18] listed down five kinds of machine learning algorithms in respect to supervised set. There are logic-based algorithms, statistical learning algorithms, instance based learning (IBL), support vector machines (SVM) and DL. The algorithms have been measured in terms of their accuracy in prediction as [19] highlighted. If the prediction produces inaccurate result, it may lead to a faulty expectation. The research is focusing on accuracy because an accurate prediction model can influence the decision making of an individual in their daily activity [20]. For example, manufacturing sector seeks help from an accurate forecast model in deciding the manufacturing rate.

Logic-based algorithms consist of decision tree (DT) and rule system. DT is claimed to produce low accuracy because of its greediness while executing the algorithm [21]. In DT, entropy and information gain play the most important role in splitting the attributes. Entropy is computing the vulnerability within the set of training because of the possibility of more than one possible splitting solution [22]. The ideal solution is having the lowest entropy that will put the probability, p either in 0 or 1 as shown in Figure 2.



$$\text{Entropy} = -0.5 \log_2 0.5 - 0.5 \log_2 0.5 = 1$$

Figure 2. Graph of entropy ft probability

Entropy is related to information gain as an accurate dt occupies lower entropy and the highest information gain. information Gain as shown in (2) is the differences in the entropy due to the partition [22].

$$\text{Gain}_{\text{split}} = \text{Entropy}(p) - \left(\sum_{i=1}^k \frac{n_i}{n} \text{Entropy}(i) \right) \quad (2)$$

Entropy (p) is the root node while k refers to splitting node and n_i stands for occurrence in partition i . It only considers the highest value that is contemplating as bias on the splitting process [23]. Although DT can handle noise and missing values very well but it provides unstable result as the whole tree need to be re-constructed when new attributes are inserted. This will take a longer execution time.

As for rule system (RS), the accuracy is poor especially for long and complex list [24]. The algorithm exists in form of $A \rightarrow B$ where A is the predecessor while B represents the successor [25]. The values returned by RS will be true or false with true is kept as valid in the system [25]. The structure of RS is easy but it can be complex when the rules are lengthy. The transparency might be affected [15]. Thus, either DT or rule system is not suitable for prediction purpose.

Statistical learning algorithms comprise of Naïve Bayes classifiers (NBC) and Bayesian networks. Those two algorithms are interconnected among each other. NBC is one of the members in the Bayesian group [26] thus they applied the same concepts in the way they work. Bayesian looks over past studies result in longer execution time [24]. The solution will be less inferential impact as the method views along on insignificant data [27]. Bayesian is also known to deal with a limited amount of continuous values [28]. The common practices are transforming the continuous attributes into discrete variables. The conversion process however able to apprehend with rough features only on the initial distribution unleashing the statistical leverages mainly if the relationship among the traits are linear [28]. Therefore, NBC and Bayesian are not competent to complete the foresee errand.

Instance-based learning (IBL) tradeoff between discovering and removing noise in data with accuracy making them to be poor [29]. It stores lots of variables include those necessary and unnecessary

ones. This issue leads to large consumption of memory spaces, higher processing time and large amount of noise data [30]. The presence of many noise data can reduce the generalization accuracy predominantly when the required instances have been reset while the noise are maintained [30]. Consequently, IBL is not well qualified for this research.

Moving on to SVM that is capable to handle classification issues and concern to regression subject [31]. SVM has an open problem with unclear result [32]. SVM comprises of support vector classification (SVC) and support vector regression (SVR) [33]. Subsequently, this paper will be looking on SVR as the research is onto regression problem. SVR draws three lines named hyperplane, upper boundary and lower boundary for generating the result. Hyperplane is sketched out according to (3) [34].

$$w \cdot x + b = 0 \quad (3)$$

Normal hyperplane is represented by w and $\frac{b}{w}$ refers to the length of space at an angle of 90 degrees between the hyperplane and the initial point [34]. The upper and lower boundary are outlined as a convex optimization issue in (4) [35].

$$\frac{1}{2} \|w\|^2 \text{ with regards to } \begin{cases} y_i - \langle w, x_i \rangle - b \leq \varepsilon \\ \langle w, x_i \rangle + b - y_i \leq \varepsilon \end{cases} \quad (4)$$

SVR redicts the final outcome by choosing the nearest support vectors (data points) to the upper and lower boundary. The challenge in SVR is working with many input variables [36]. The accuracy is declining towards the growing size of input variables [37]. The data points are dispersed when fit with the network model. SVR only considers those near to the boundaries without acknowledging the sequence disrupting the accuracy. Accordingly, SVM is not applicable in this case.

The last algorithm is deep learning derived from neural network which able to mimic human action on learning by example on its own. Neural network is similar to brain that full of science fiction of connotations of the Frankenstein mythos [38]. Neural network is a machine learning algorithm and the base for the creation of deep learning [39]. Basically, the basic blocks of deep learning came from neural network to build up the deep neural network.

RNN is classified as deep neural network due to its several layers of hidden states which have the capability to solve the non-linear problem. The hidden layers act as memory to store the previous information [40]. RNN consists the same cycle as Hopfield network and long short-term memory (LSTM) since that it can loop back the process to initial point. RNN modelizes dynamic structure on yielding the output through the observation on current and past information [41]. It is well-known in sequencing data thus the hidden state will be updated once new data is input [42]. However, RNN suffers from exploding and vanishing gradients [41]. Gradient involves the changing of output while input is modified. The value of gradient can turn into smaller or bigger ones upon the feedback loops of BPTT result in unstable error [43]. Hence, RNN confronts with short term memory [44].

GRU solves the problem with its two gates named update and reset to manage the succession of data in the time of learning them for predicting through the information organized in a long chain [45]. Update gate receives the input and sort them out before hand over to memory [42]. ast state will remained if the update gate is located near to one. In contrast to reset gate close by one, it will work together with RNN. If the gate reaches zero, hidden layer is the result of MLP and the present ones will be assigned as input. Figure 3 shows the structure of GRU.

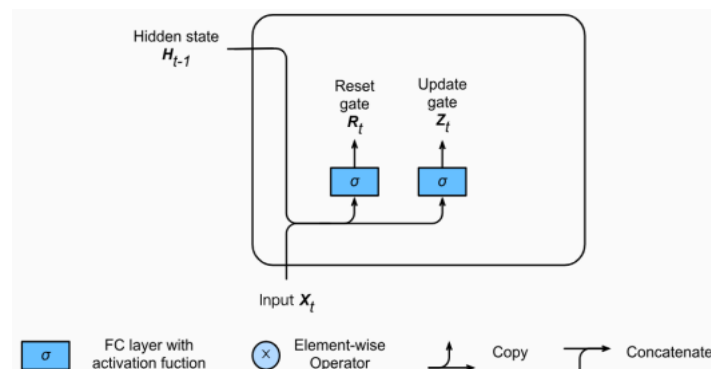


Figure 3. RNN-GRU architecture

RNN-GRU is proved to execute faster around 18.16% than RNN-LSTM when median values are combined together [46]. This means that GRU can work faster than LSTM without compromising the accuracy. One of the existing literature on time series forecasting using GRU model is written by Rahman *et al.*, [47] it is used to predict the price of stock market. The authors started fetch real time dataset gathered from Yahoo Finance [47]. Pandas collections are used to separate the data into training and testing [47]. The current input is marked as t while the output will be classified into $t+1$ [47]. Then, the data is scaled and reshape into appropriate composition. Since that GRU is associated with neural network (NN), all elements belong to NN is activated [47]. The proposed method is fit into training data and output will be seeing through testing. Real values are set as the benchmark for comparison between training and testing. The result is visualized for further evaluation. To conclude, the first data came from lower's company (LOW) with RMSE of 0.0127464 followed by Coca-Cola Company at 0.0144508. The third company named Apple Incorporation displays 0.013996 as the RMSE. This demonstrates the excellency of GRU with time series data for prediction activity.

The statement is supported by Le *et al.*, they claimed that GRU can deliver equivalent great outcome with simpler structure [48]. The writers utilized 18 years of water level data at the Can Tho river from January 2000 to January 2018. The data is split up into three division categorized as training, testing and validation. Seventeen years from 2000 to 2016 are allocated for training [48]. Year of 2017 are used for validation and five days in January 2018 granted for prediction and evaluation. The model is trained to identify the value for its parameters [48]. This research only required one variable of data named water level which make the time step to be one as the dimension for input and output [48]. The hidden layers are decided to be in range 5 to 20. The design is finalized with number of epochs, dropout, learning rate and Adam algorithm towards the data. As the result, MAE is tabulated among 0.087 to 0.106. Hence, the authors able to prove that GRU can yield not much significant of error values with fewer gates.

2. RESEARCH METHOD

Research activity for this paper is presented here to satisfy the research objectives. There are seven essential steps in the research activities as demonstrated in Figure 4.

- Collection of data

Oil and gas organizations support this research by giving out its historical of one year data. The size is estimated at 55 GB.

- Data cleaning

There are three columns of name of machines or tags, time and the value of tags. A meeting is conducted with a data expert from the involved company to consult on the data. Any noise or error is resolved amid the session.

- Division of data

The ratio set for training and testing is 7:3.

- RNN-GRU construction

The time series data is feed towards the algorithm. The values are normalized into 0 and 1. GRU is adjusted to fit with data structure. A random value is set at number of epochs and look back. The error is computed through RMSE during loop back with ADAM optimizer to update the network weight.

- Training of RNN-GRU

Update gate in charge of taking in input to figure out the amount of required information to along with the content of information.

$$z_t = \sigma(W^{(z)}x_t + U^{(z)}h_{t-1}) \quad (5)$$

x_t represents the present input is multiplied with its weight $W^{(z)}$ in the meantime of multiplication of h_{t-1} , past data with its input $U^{(z)}$. Both answers are aggregated and times with sigmoid function. The result is displayed as z_t in the range of 0 to 1.

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot h_t \quad (6)$$

z_t and h_{t-1} is element-wise and added up with element-wise of $1 - z_t$ and h_t . h_t is referred to the essential information to be stored in memory will be passing down to reset gate.

$$r_t = \sigma(W^{(r)}x_t + U^{(r)}h_{t-1}) \quad (7)$$

At the reset gate, x_t is again multiplied to the weight $W^{(r)}$ and sum up with the result of h_{t-1} times to weight of $U^{(r)}$. Sigmoid is applied to scale the answer between 0 to 1. In (7) is calculating the amount of data to be removed. Other than resetting the unnecessary data, reset also issued memory spaces to keep the result of (5) and (6) as demonstrated in (8).

$$h_t = \tanh(Wx_t + r_t \odot Uh_{t-1}) \quad (8)$$

- i. First, x_t is times with weight, W and weight, U is multiplied with h_{t-1} .
- ii. Hadamard function is used upon r_t and Uh_{t-1} to reset the non-essential data.
- iii. Last, (i) and (ii) is total up and times with tanh non-linear activation function.

Weight of network is updated through ADAM optimizer and error will be exhibited on RMSE. Training is iterated based on the value set at number of epochs.

– Testing of RNN-GRU

The error addressed at this phase is evaluated with previous research. A graph consisting actual data with predicted data points is visualized.

– Validation of RNN-GRU

Re-testing on RNN-GRU is performed to compare the deviation with the result of previous research on RNN-GRU.

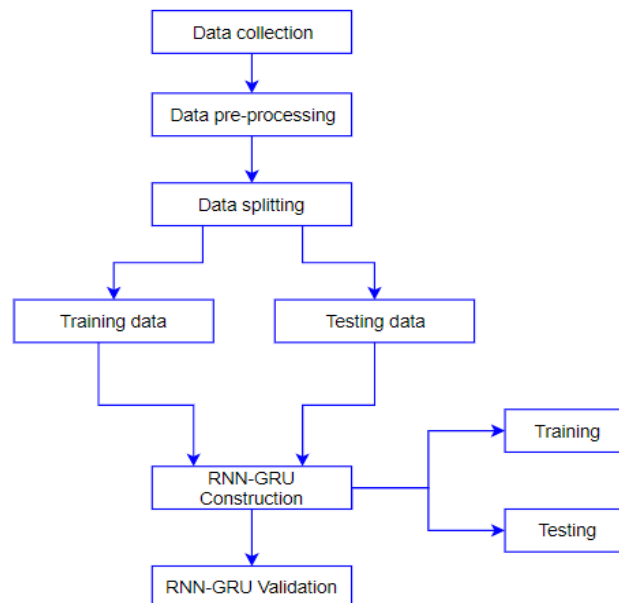


Figure 4. Research activities

3. RESULTS AND DISCUSSION

The result of RNN-GRU is showed here with the comparison between RNN-LSTM.

3.1. Proof of concepts

The experiment is performed to:

- To prove that RNN-GRU can process big data size within a limited time allocated.
- To prove that RNN-GRU can forecast time series data and produce high accuracy.

3.2. Results and discussion

The prediction results generated by RNN-GRU are tabulated. The parameters of GRU are assigned to default values.

- Number of neurons=2, 4, 6
- Number of look back=3
- Number of epochs=10
- Number of batch size=1

Figure 5 exhibits a plotted graph of sensor values on equipment in oil and gas sector feat with data size. There are three lines indicating blue for the actual data recorded from sensor, testing in orange and green as testing phase. Those lines need to be between 25 kpa (sensor measurement) to 400 kpa. Any of them that go beyond 400 kpa is forecasted to fail in contrast to error that occur if the data points went below than 25 kpa.

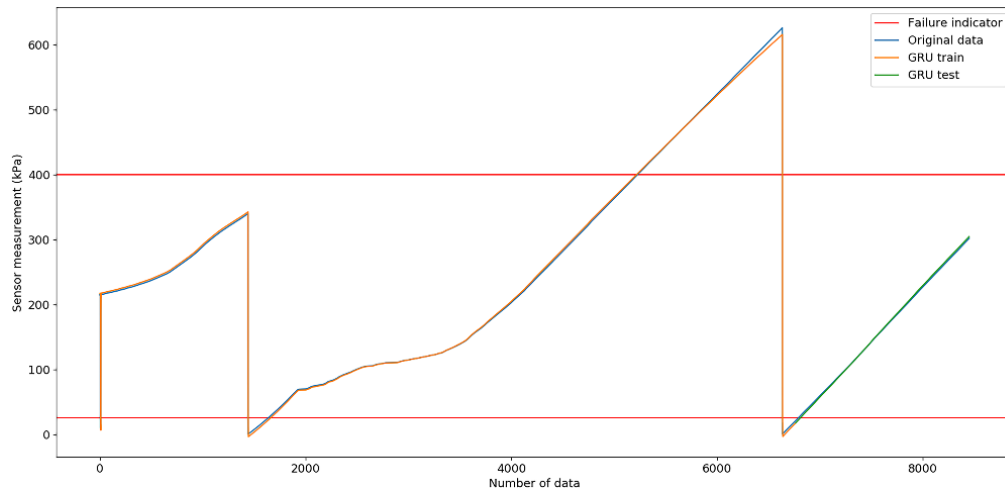


Figure 5. Graph of predicted time series data with initial values

Table 1 signifies the performance of RNN-GRU that include value of RMSE with time execution for 10 000 rows of data. The highest accuracy of testing and training are 13.13 and 4.58 as number of neurons at six. When the number of neurons is four, RMSE of testing is 13.35. Training is equal to 4.69. This means that the accuracy decline along with the amount of neurons. The RMSE of training and testing for two number of neurons are 13.93, 4.94. Thus, the parameters that surrounded RNN-GRU can influence its performance as the higher the number of neurons can improve the accuracy.

Table 1. Performance of RNN-GRU with distinct of parameters values

Number of epochs	Number of look back	Number of neurons	RMSE		Time (Min)
			Training	Testing	
10	3	2	4.94	13.93	2.98
		4	4.69	13.35	3.08
		6	4.58	13.13	3.13

In order to prove that RNN-GRU can run faster than RNN-LSTM with no significant in accuracy produced thus the same set of data is tested with RNN-LSTM. The result is tabulated in Table 2. Two number of neurons result in 5.01 and 14.03 for RMSE of training and testing. The training RMSE is decrease to 13.10, with 4.50 for training when number of neurons is four. Six neurons produced 4.71 of training and 12.55 for testing.

Table 2. Performance of RNN-LSTM with distinct of parameters values

Number of epochs	Number of look back	Number of neurons	RMSE		Time (Min)
			Training	Testing	
10	3	2	5.01	14.03	3.00
		4	4.50	13.10	3.13
		6	4.71	12.55	3.17

The differences that want to be highlighted here is the time of execution. Based on both Tables 1-2, it is proven that GRU can run faster than LSTM. This is because GRU consists of less one gate than LSTM yet the accuracy can be claimed with not many differences.

4. CONCLUSION

Time series data is not an easy task to handle with due to the update of data in each second. This issue leads towards emergence of big data problem. The existence of machine learning (ML) algorithm helps in managing the flow of data to forecasting action. The forecasting task plays an important role in many sectors by monitoring the state of machines involved while producing product. Otherwise, the organization need to carry much losses when experiencing sudden breakdown. Subsequently, this paper illustrates recurrent neural network-gated recurrent unit (RNN-GRU) for time series forecasting of machine condition through dataset obtained from oil and gas company. The technique comprises with lots of parameters such as learning rate, number of epochs, number of neurons in hidden layer, batch size and many more. The values assigned to them affects the performance of the method. An experiment of input time series data into RNN-GRU model to forecast the reliability of machine while proving to produce low RMSE. GRU is clarified to have similar accuracy with recurrent neural network-long short term memory (RNN-LSTM) but it can finish processing the data faster. The investigation has inferred that RNN-GRU can produce up to 87% of accuracy.

REFERENCES

- [1] R. M. Aliguliyev and Y. N. Imamverdiyev, "Big data strategy for the oil and gas industry : General," *Problems of Information Technology*, no. 2, pp. 31-42, 2017.
- [2] Jan Grandell, "Time series analysis.," Lecture Notes, KTH, Sweden, 2000.
- [3] I. Velchamy, U. Ilango, and N. Ramesh, "Time series data mining research problem, issues, models, trends and tools," *International Journal of Computer Engineering & Applications*, vol. 9, no. 5, pp. 59-75, 2015.
- [4] S. P. Singh and U. C. Jaiswal, "Machine learning for big data : A new perspective," *International Journal of Applied Engineering Research*, vol. 13, no. 5, pp. 2753-2762, 2018.
- [5] A. Dogan and D. Birant, "Machine learning and data mining in manufacturing," *Expert System with Applications*, vol. 166, p. 114060, 2020.
- [6] S. L. Nita, "Machine learning techniques used in big data," *Scientific Bulletin of Naval Academy*, vol. 19, no. 1, pp. 466-471, 2016.
- [7] A. Agrawal, J. Gans, and A. Goldfarb, "Prediction machines: The simple economics of artificial intelligence," Harvard Business Review Press, 2018.
- [8] J. Nakamura, "Predicting time to failure of industrial machines with temporal data mining," Dissertation, Masters of Science, University of Washington, 2007.
- [9] G. Immerman, "The real cost of downtime in manufacturing," [Online]. Available at: <https://www.machinemetrics.com/blog/the-real-cost-of-downtime-in-manufacturing>. [Accessed: 15-Apr-2019].
- [10] InSource Solution, "The total cost of downtime in industrial automation," [Online]. Available at: http://go.insource.solutions/Stratus-Nurture_03.html.
- [11] B. M. Pavlyshenko, "Machine-learning models for sales time series forecasting," *Data*, vol. 4, no. 1, pp. 1-11, 2019.
- [12] J. F. Torres, A. Troncoso, I. Koprinska, Z. Wang, and F. Martínez-Álvarez, "Deep learning for big data time series forecasting applied to solar power," *The 13th International Conference on Soft Computing Models in Industrial and Environmental Applications*, pp. 123-133, 2018.
- [13] G. Rebala, A. Ravi, and S. Churiwala, "An introduction to machine learning," Springer Nature Switzerland, 2019.
- [14] M. Praveena and V. Jaiganesh, "A literature review on supervised machine learning algorithms and boosting process," *International Journal of Computer Applications*, vol. 169, no. 8, pp. 32-35, 2017.
- [15] J. Hurwitz and D. Kirsch, "Machine learning for dummies a wiley brand," John Wiley & Sons, Inc, 2018.
- [16] A. Azran and Z. Ghahramani, "A new approach to data driven clustering," *Proceedings of the 23rd International conference on Machine Learning*, pp. 57-64, 2006.
- [17] G. Shobha and S. Rangaswamy, "Machine learning," *Handbook of Statistics*, vol. 38, pp. 197-228, 2018.
- [18] M. Iqbal and Z. Yan, "Supervised machine learning approaches: A survey," *International Journal of Soft Computing*, vol. 5, no. 3, pp. 946-952, 2016.
- [19] J. Frost, "Understand precision in predictive analytics to avoid costly mistakes - statistics by Jim," Regression Analysis: An Intuitive Guide for Using and Interpreting Linear Model. [Online]. Available at: <https://statisticsbyjim.com/regression/prediction-precision-applied-regression/>. [Accessed: 10-Apr-2019].
- [20] S. Wang and W. Chaovaitwongse, "Evaluating and comparing forecasting models," *Wiley Encyclopedia of Operations Research and Management Science*, 2010.
- [21] L. Rokach and O. Maimon, "Data mining and knowledge discovery handbook: Chap. 9," Springer, pp. 165-192, 2010.
- [22] M. Singh, S. Sharma, and A. Kaur, "Performance analysis of decision trees," *International Journal of Computer Applications*, vol. 71, no. 19, pp. 10-14, 2013.
- [23] B. Baykara, "Impact of evaluation methods on decision tree accuracy," Thesis, School of Information Sciences, Computer Science, University of Tampere, 2015.
- [24] L. Lengyel, "Validating rule-based algorithms," *Acta Polytechnica Hungarica*, vol. 12, no. 4, pp. 59-75, 2015.
- [25] C. Anantaram, G. Nagaraja, and K. V. Nori, "Verification of accuracy of rules in a rule based system," *Data & Knowledge Engineering*, vol. 27, no. 2, pp. 115-138, 1998.

- [26] J. Brownlee, "A tour of machine learning algorithms," [Online]. Available at: <https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/>. [Accessed: 10-Feb-2019].
- [27] B. Efron, "Frequentist accuracy of bayesian estimates," *Journal of the Royal Statistical Society. Series B, Statistical Methodology*, vol. 77, no.3, pp. 617, 2015.
- [28] L. Uusitalo, "Advantages and challenges of Bayesian networks in environmental modelling," *Ecological Modelling*, vol. 203, no. 3-4, pp. 312-318, 2007.
- [29] D. W. Aha, D. Kibler, and M. K. Albert, "Instance-based learning algorithms," vol. 6, no. 1, pp. 37-66, 1991.
- [30] D. R. Wilson and T. R. Martinez, "Reduction techniques for instance-based learning algorithms," *Machine Learning*, vol. 38, no. 3, pp. 257-286, 2000.
- [31] E. A. Zanaty and S. Aljahdali, "Improving the accuracy of support vector machines," *Computers and Their Applications*, pp. 196-202, 2008.
- [32] L. Auria and R. A. Moro, "Support vector machines (SVM) as a technique for solvency analysis," *DIW Berlin Discussion Paper*, no. 811, 2008.
- [33] D. Basak, S. Pal, and D. C. Patranabis, "Support vector regression," *Neural Information Processing-Letters and Reviews*, vol. 11, no. 10, pp. 203-224, 2007.
- [34] T. Fletcher, "Support vector machines explained," Tutorial Paper, pp. 1-19, 2009.
- [35] A. J. Smola and B. Scholkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, no. 3, pp. 199-222, 2004.
- [36] A. Corazza, S. Di Martino, F. Ferrucci, C. Gravino, F. Sarro, and E. Mendes, "Using tabu search to configure support vector regression for effort estimation," *Empirical Software Engineering*, vol. 18, no. 3, pp. 506-546, 2013.
- [37] 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 Journal of Civil Engineering*, vol. 18, no. 4, pp. 941-948, 2014.
- [38] R. M. Cesar and L. da Fontoura Costa, "An introduction to neural networks," *Nemocomputing*, vol. 14, no. 1, pp. 101-104, 2002.
- [39] Md Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, Paheding Sidike, Mst S. Nasrin, Mahmudul Hasan, B. C. V. Essen, A. A. S. Awwal, and V. K. Asari, "A state-of-the-art survey on deep learning theory and architectures," *Electronics*, vol. 8, no. 3, p. 292, 2019.
- [40] H. Salehinejad, S. Sankar, J. Barfett, E. Colak, and S. Valaee, "Recent advances in recurrent neural networks," *ArXiv Preprint:1801.01078*, 2017.
- [41] G. Chen, "A gentle tutorial of recurrent neural network with error backpropagation," *ArXiv Preprint:1610.02583*, 2016.
- [42] J. Chung, K. Kastner, L. Dinh, K. Goel, A. Courville, and Y. Bengio, "A recurrent latent variable model for sequential data," *Advances in Neural Information Processing Systems*, pp. 2980-2988, 2015.
- [43] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," *International Conference on Machine Learning*, pp. 1310-1318, 2013.
- [44] S. Hochreiter, "The vanishing gradient problem during learning," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 6, no. 2, pp. 107-116, 1998.
- [45] J. Yuan and Y. Tian, "An intelligent fault diagnosis method using GRU neural network towards sequential data in dynamic processes," *Processes*, vol. 7, no. 3, pp. 152-172, 2019.
- [46] R. Rana, J. Epps, R. Jurdak, X. Li, R. Goecke, M. Breretonk, and J. Soar, "Gated recurrent unit (GRU) for emotion classification from noisy speech," *ArXiv preprint*, pp. 1-9, 2016.
- [47] M. O. Rahman, Md. S. Hossain, T-S. Junaid, Md. S. A. Forhad, and M. K. Hossen, "Predicting prices of stock market using gated recurrent units (GRUs) neural networks," *International Journal of Computer Science and Network Security*, vol. 19, no. 1, pp. 213-222, 2019.
- [48] X-H. Le, H. V. Ho, and G. Lee, "Application of Gated Recurrent Unit (GRU) network for forecasting river water levels affected by tides," *International Conference on Asian and Pacific Coasts*, pp. 673-680, 2019.