E3S Web of Conferences **501**, 01023 (2024) https://doi.org/10.1051/e3sconf/202450101023
ICCSEI 2023

Bidirectional Long Short-Term Memory (Bi-LSTM) Hourly Energy Forecasting

Aji Prasetya Wibawa[1*], *Akhmad* Fanny Fadhilla[1], *Andien* Khansa'a Iffat Paramarta[1], *Alfiansyah* Putra Pertama Triono[1], *Faradini* Usha Setyaputri[1], *Ade* Kurnia Ganesh Akbari[1], and *Agung* Bella Putra Utama[1]

[1]Department of Electrical Engineering and Informatics, Faculty of Engineering, Universitas Negeri Malang, Malang 65145, Indonesia

Abstract. The growing demand for energy, especially in urban and densely populated areas, has driven the need for smarter and more efficient approaches to energy resource management. One of the main challenges in energy management is fluctuations in energy demand and production. To overcome this challenge, accurate and careful forecasting of hourly energy fluctuations is required. One method that has proven effective in time series forecasting is using deep learning. The research phase uses the CRISP-DM data mining methodology as a common problem solver for business and research. The scenarios tested in the study used 5 attribute selection scenarios based on correlation values based on target attributes and 2 normalization scenarios. Then, the deep learning model used is Bi-LSTM with hyperparameter tuning grid search. Performance measurement evaluation is performed with MAPE, RMSE, and R2. Based on the tests conducted, it was found that the Bi-LSTM model produced the best MAPE of 7.7256%. RMSE of 0.1234. and R2 of 0.6151 at min-max normalization. In comparison, the results on the z-score normalization are lower with the best MAPE value produced at 10.5525%. RMSE of 0.7627. and R2 of 0.4186.

1 Introduction

In this modern era, electrical energy has become a crucial component in the daily life and economic sustainability of a country [1]. The growing demand for energy, especially in urban and densely populated areas, has driven the need for smarter and more efficient approaches to energy resource management [2]. The availability of reliable and efficient electrical energy is essential to maintain the activities of the industrial [3], commercial [4], and domestic sectors [5]. One of the main challenges in energy management is fluctuations in the demand and production of electrical energy, especially on short-time scales such as hourly fluctuations. To overcome this challenge, accurate and careful forecasting of hourly energy fluctuations is required. By having accurate forecasting of energy fluctuations, power system operators can plan energy operations and production more efficiently. Industries can regulate the use of different energy sources, including renewables, based on better demand predictions.

Analysis of energy forecasting data in the form of time series can be done using deep learning methods because it effectively overcomes the complexity of the data [6][7]. Deep learning is a branch of machine learning neural networks that consists of many hierarchically connected layers, ranging from the input layer to the output layer [8]. Some deep learning methods that have been developed to perform time series data analysis include Recurrent Neural Network (RNN) [9], Convolutional Neural Network (CNN) [10], Long Short-term Memory (LSTM) [11], Gated Recurrent Unit (GRU) [12].

Multivariate time series data have several attributes that have different scales [13]. It is necessary to rearrange the attribute scale to have a more standardized or normal distribution of values. This problem can be avoided by normalizing to get data with the same range of values [14]. Some commonly used normalization methods in deep learning models include Minmax [15], Z-score [16], RobustScaler [17], and MaxAbsScaler [18]. The selection of normalization methods is based on data characteristics and research focus so as to effectively produce deep learning models with optimal performance.

Based on the description that has been given by previous researchers. This study focuses on normalization selection on multivariate time series data analysis using the Bi-LSTM method. All attributes in the data will be used with several scenarios to obtain the best accuracy. Evaluate performance measurement using Mean Absolute Percentage Error (MAPE), Root Mean Square Error (RMSE), and R-Squared (R2) values.

* Corresponding author: aji.prasetya.ft@um.ac.id

 https://doi.org/10.1051/e3sconf/202450101023

2 Literature Review

2.1 Normalization

Normalization is a technique of operating on a dataset to change the value, thereby, the dataset has the same value scale with no dominating features [19]. Normalization becomes a foundational component in machine learning that is effective for improving model performance [20]] because it accelerates deep learning training and acts as a form of generalizing data [17].

2.2 Deep Learning

Deep learning is a branch of machine learning that uses the architecture of deep neural networks to learn complex patterns from data [21]. Deep learning focuses on developing and training models that can automatically learn in-depth feature representations of data without the need for manually generated features. Deep learning architecture consists of many hierarchically connected layers, from the input layer to the output layer [22]. Approaches using deep learning can be used to analyze multivariate time series data.

2.3 Bi-LSTM

Bi-LSTM has been used in various fields, including sea level prediction [10][23] prediction [11][12][13][24][25][26]. In its implementation, Bi-LSTM has advantages and disadvantages.

The advantage of Bi-LSTM lies in its ability to improve model performance in studying time-series data over a long period of time. In addition, the way Bi-LSTM works using information from 2 directions can be utilized to improve the accuracy of a model [27]. Bi-LSTM also has the advantage of predicting non-linear time-series data and has noise [25]. However, Bi-LSTM has some drawbacks.

The weakness of Bi-LSTM is in its complex structure, where there is an addition of LSTM backward layer so that the operation works slowly and takes a long time [25]. In addition, the bidirectional use of Bi-LSTM requires more computing and memory than LSTM [28]. Therefore, this model must also be matched with needs. The process flow of the Bi-LSTM algorithm is presented in Figure 1.

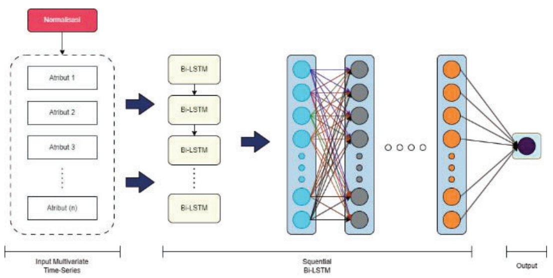

Fig. 1. Bi-LSTM Process Flow.

From Figure 1, the process flow of the Bi-LSTM algorithm in the context of time series data begins with preparing the data in the appropriate format. The inner BiLSTM layer consists of two components, namely forward and reverse LSTM. The sequence of data entered into the forward LSTM represents information from the past to the present, while the sequence of data entered into the backward LSTM represents information from the present to the future. This process allows the network to capture temporal patterns that exist in the time series. After the entire sequence in the time window is processed, the information from both LSTM directions is combined. The result of this step results in a richer representation of that time window. The next step is to direct that representation to an output layer that matches the purpose of time series modeling. For example, if the goal is prediction, the output layer might be a single neuron that generates a predictive value for the next point in time. If the goal is time classification, the output layer will probably have several neurons according to the number of possible classes.

3 Research Methods

This research used the Cross Industry Standard Process for Data Mining (CRISP-DM) data mining methodology as a common problem solver for business and research [29]. The CRISP-DM methodology consists of six stages, namely Business Understanding, Data Understanding, Data Preparation, Modelling, Evaluation, and Deployment [30], as illustrated in Figure 2.

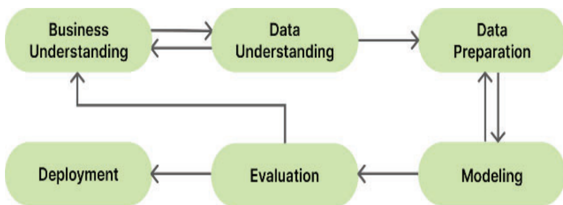

Fig. 2. Research Phase.

3.1 Business Understanding

Business Understanding is an initial stage of assessing research needs and objectives [31]. Multivariate time series data has several features with different value scales. Therefore, a process is needed to reset the scale of values on the data with normalization. Therefore, this study aims to find normalization methods that match the Bidirectional Long Short-term Memory (Bi-LSTM) deep learning model. The normalization methods used in this study were Min-max and Z-Score.

3.2 Data Understanding

Data understanding contains an analysis of the understanding of the collected data [32]. The dataset used in this study was sourced from the Kaggle site by downloading the file.csv. There were 29 attributes in the dataset and 35064 instances, which were later selected to select features. The attributes that were used in this study included total actual load. The details of the attributes are shown in Table 1.

Table 1. Dataset Attribute Details.

Attribute	Data type	Description (min, max)
Generation biomass	Float	(0, 592)
Generation fossil brown coal/lignite	Float	(0, 999)
Generation fossil coal/derived gas	Float	(0)
Generation fossil gas	Float	(0, 20034)
Generation fossil hard coal	Float	(0, 8359)
Generation fossil oil	Float	(0, 449)
Generation fossil oil shale	Float	(0)
Generation fossil peat	Float	(0)
Generation geothermal	Float	(0)
Generation hydro pumped storage aggregated	Float	NaN
Generation hydro pumped storage consumption	Float	(0, 4523)
Generation hydro run-off river and poundage	Float	(0, 2000)
Generation hydro water reservoir	Float	(0, 9728)
Generation marine	Float	(0)
Generation nuclear	Float	(0, 7117)
Generation other	Float	(0, 106)
Generation other renewable	Float	(0, 119)
Generation solar	Float	(0, 5792)
Generation waste	Float	(0, 357)
Generation wind offshore	Float	(0)
Generation wind onshore	Float	(0, 17436)
Forecast solar day ahead	Float	(0, 5836)
Forecast wind offshore day ahead	Float	NaN
Forecast wind onshore day ahead	Float	(0, 17430)
Total load forecast	Float	(0, 41390)
Total load actual	Float	(0, 41015)
Price day ahead	Float	(0, 98.69)
Price actual	Float	(0, 99.95)

3.3 Data Preparation

The data preparation stage before the classification process includes data preparation [33]. Generally used techniques for data preparation include data cleaning, data integration, data transformation, data reduction, and data discretization [34]. In this study, the technique used was missing value handling. Missing values can cause problems in the case of machine learning and statistical models because models can produce inaccurate results [[25,26]. This research handled missing values by deletion. The data used contained 2 columns of NaN. The 2 columns were deleted because they had no correlation with the target attribute.

3.4 Modelling

Modeling is further related to the design of the research model used [35]. The design of the research model is shown in Figure 3.

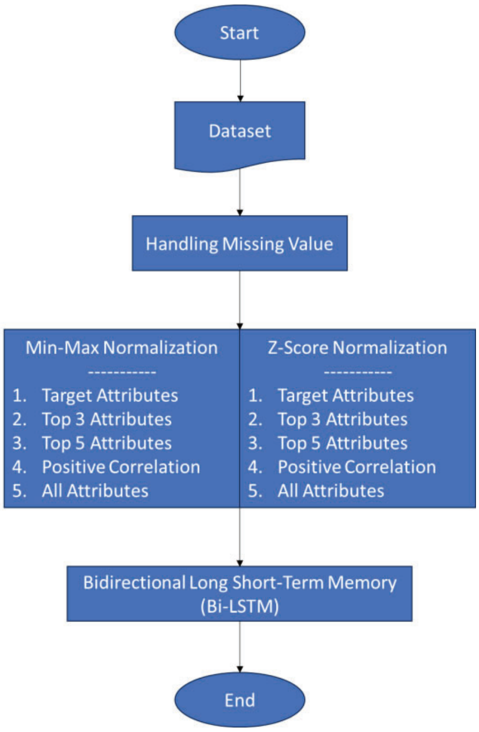

Fig. 3. Research Modeling Framework.

The model created using the Bi-LSTM method uses 2 normalization scenarios. Normalization is a technique of operating on a dataset to change the value so that the dataset has the same value scale and there are no dominating features [19]. Min-max normalization transforms attributes or variables in the dataset into values between 0 and 1. This transformation process can be performed on all features in the data so that features have values in the range of 0 to 1. The process of calculating the normalization of Min-max as in (1).

$$x' = \frac{(x - x_{min})}{(x_{max} - x_{min})} \quad (1)$$

Z-score normalization, known as standardization, is a technique in which the value of the attribute will be normalized based on the mean (μ) and standard deviation (σ). The process of calculating the normalized Z-score as in (2).

$$x' = \frac{x - \mu}{\sigma} \quad (2)$$

After the data was preprocessed, the following process was modeled by creating a model with the Bi-LSTM method. The parameters used in making this model were obtained from the hyperparameter tuning grid search.

Processing a large number of instances leads to longer training times. Therefore, the batch size was adjusted to 100 and 1000 sizes for faster processing. It also relates to the number of epochs used. Epochs that are too small can cause models not to reach convergence. The use of hidden layers is limited to a maximum of 10 and neurons numbered 32 and 64 to minimize the risk of overfitting the model. The loss function was chosen because the three functions are loss functions for regression, according to the model created.

Adam and RMSprop were chosen as optimizers because they resulted in minimal loss compared to other optimizers.

Model testing began by training data to improve model performance. The results of the training were used to test the performance of the model in data testing. The comparison of training and testing data was 100:100 for maximum data usage.

3.5 Evaluation

The evaluation stage was carried out by evaluating the results of forecasting. It aimed to determine the accuracy of the model that had been carried out by calculating the error value. The evaluations used were MAPE, RMSE, and R2. MAPE & RMSE functions were used to determine the accuracy of the performance of the model that had been created. The experiment was conducted 5 times in each scenario and then averaged. The equation of the functions MAPE, RMSE, and R2 [36] as in (3) to (5).

$$\mathrm{MAPE} = \frac{1}{m}\sum_{i=1}^{m}\frac{|Y_i - X_i|}{Y_i} \quad (3)$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(X_i - Y_i)^2} \quad (4)$$

$$R^2 = 1 - \frac{\sum_{i=1}^{m}(X_i - Y_i)^2}{\sum_{i=1}^{m}(\bar{Y} - Y_i)^2} \quad (5)$$

3.6 Deployment

The deployment phase was carried out by reporting the results of the model evaluation. Evaluation analysis was performed by comparing MAPE, RMSE, and R2 values. The resulting MAPE & RMSE values closer to 0 indicate a more accurate prediction model. Conversely, if the resulting value is higher, it indicates that the prediction model is increasingly inaccurate.

4 Results and Discussion

This chapter discusses the next stages, which include Data Preparation, Modeling, Evaluation, and Deployment.

4.1 Data Preparation

The data preparation stage carried out was missing value handling. There were 2 attributes removed because they had NaN values, namely Generation hydro pumped storage aggregated and Forecast wind offshore day ahead. These attributes were removed because missing values cause sampling bias. In addition to the attribute with the NaN value, there is a comparison attribute to the predicted value generated by the model, namely the total load forecast. The attribute was excluded from processing because it is a predictive value for the target attribute of the Total actual load. These attributes can be used to compare prediction values from models created with those used in industry. The removal of these attributes causes the number of attributes to decrease from 29 to 26 attributes.

4.2 Modelling

The attribute selection process was carried out by calculating the correlation of each attribute with the target attribute, namely 'total actual load.' The correlation used in this study was Pearson. This type of correlation was used because the target attribute has a normal distribution where the data has no significant differences and no outliers. The results of the correlation value calculation were sorted in descending (large to small) for attribute selection. The attribute selection process was adjusted to 5 predefined scenarios. The next process was to carry out the normalization process with 2 normalization methods, namely Min-max and Z-score.

After the data preprocessing stage, the next stage was the creation of the Bi-LSTM model. The created model requires parameters that correspond to the characteristics of the data. The determination of these parameters was completed by hyperparameter tuning grid search in order to get the best combination of hyperparameters and the model provides optimal performance. In addition, other hyperparameters, such as the activation function, were determined using tanh because it was centered on point 0 in order to achieve convergence faster and dropout by 0.2 to avoid the risk of overfitting. The results of the hyperparameter tuning grid search are shown in Table 2.

Table 2. Hyperparameter Tuning.

Parameter	Search Space	Result
Batch Size	‘100’, ‘1000’	100
Epoch	’50’, ‘100’	50
Hidden Layer	‘2’, ‘5’, ‘10’	2
Loss Function	‘mse’, ‘mae’, ’huberloss’	MSE
Neuron	‘32’, ‘64’	32
Optimizer	‘adam’, ‘rmsprop’	Rmsprop

Testing using parameters obtained from hyperparameter tuning results in a Bi-LSTM model that can achieve convergent conditions. The convergence graph of the Bi-LSTM model with min-max and z-score normalization is shown in Figure 4 and Figure 5.

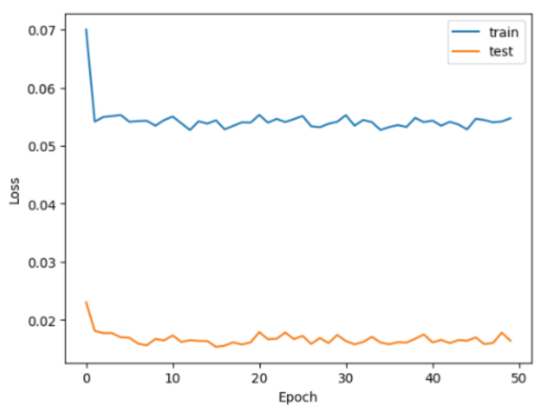

Fig. 4. Min-max Convergence Chart.

https://doi.org/10.1051/e3sconf/202450101023

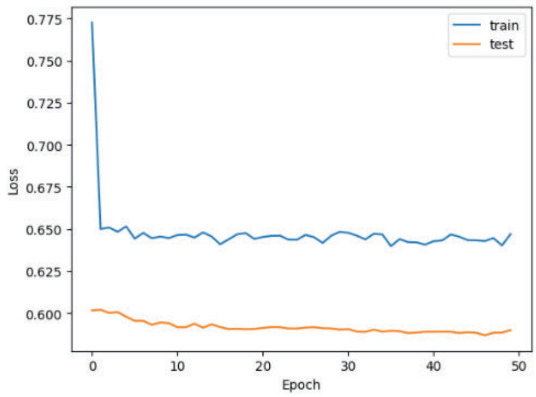

Fig. 5. Z-score Convergence Graph.

4.3 Evaluation

The model evaluation methods used were MAPE, RMSE, and R2. The created Bi-LSTM model was run 5 times and produced an evaluation value of 5 per scenario. These values were on average, for producing the final values of MAPE, RMSE, and R2 in each scenario. The evaluation results are shown in Table 3.

Table 3. Experimental Results.

Normalization	Scenario	MAPE (%)	RMSE	R^2
Min-max	Target Attributes	8.6370	0.13442	0.5440
	Top 3 Attributes	8.2884	0.13080	0.5680
	Top 5 Attributes	8.1708	0.12956	0.5762
	Positive correlation	7.7256	0.12346	0.6151
	All Attributes	7.8193	0.12520	0.6042
Z-score	Target Attributes	10.7642	0.7687	0.4088
	Top 3 Attributes	10.8853	0.7719	0.4035
	Top 5 Attributes	10.6842	0.7676	0.4100
	Positive correlation	10.5710	0.7633	0.4186
	All Attributes	10.5525	0.7627	0.4175

Based on the results in Table 3, min-max normalization produces the best evaluation value in attribute scenarios without negative correlation and NaN. This scenario produces the best MAPE value with a value of 7.7256, an RMSE value of 0.12346, and an R2 value of 0.6151.

4.4 Deployment

Deployment was the last stage in CRISP-DM, which contained a report on the results at the evaluation stage. The selection of the best scenario was completed based on the MAPE value in each normalization method. The MAPE value was used as a reference to measure model performance by calculating the percentage of prediction and actual value errors. This is because the MAPE value is a percentage, thereby, facilitating easier understanding. The best normalization is determined by normalization, resulting in low MAPE and RMSE values and high R2 values. The evaluation results of MAPE, RMSE and R2 are presented in Figure 6 to Figure 8.

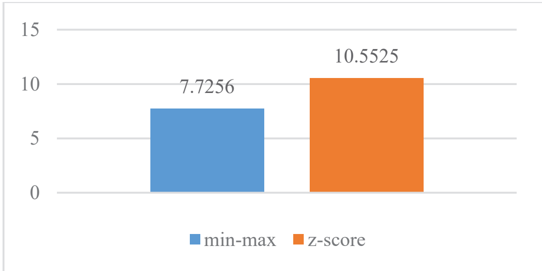

Fig. 6. MAPE Evaluation.

Figure 6 shows that the MAPE value produced by the min-max normalization of 7.7256% is lower than the MAPE value of 10.5525% produced by the z-score normalization.

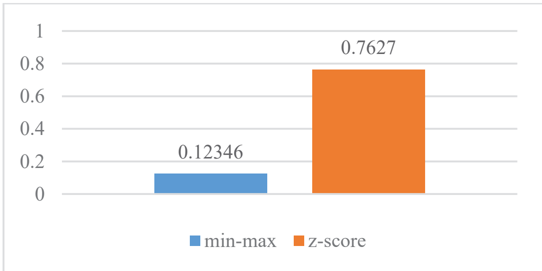

Fig. 7. RMSE Evaluation.

In addition, Figure 7 shows that the RMSE value of 0.12346 generated by the scenario with the min-max normalization method is lower than the RMSE value of 0.7627 generated by z-score normalization.

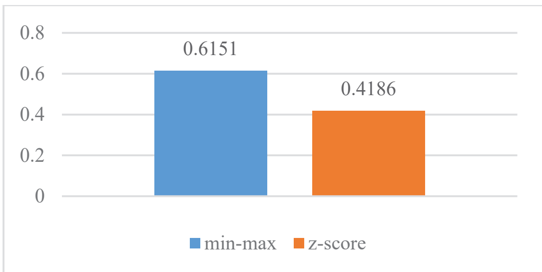

Fig. 8. R2 Evaluation.

Figure 3 illustrates that the R2 value of 0.6151 generated by the scenario with the min-max normalization method is higher compared to the R2 value of 0.4186 generated by the z-score normalization.

Following the aforementioned results, the min-max normalization method produces better performance than the z-score. Overall, each evaluation method performed well at min-max normalization. Therefore, in this study, the Bi-LSTM model produced optimal performance using the min-max normalization method. These results are in line with previous studies [37], which described better RMSE results on the min-max normalization method of 9.99, with a z-score was 10.50. However, another study [38] revealed that models using Z-score normalization provide higher accuracy than models using min-max normalization.

The difference in results can be caused by differences in the characteristics of the data used. In addition to differences in data characteristics, the use of the Tanh activation function may also present an effect. The activation function continues information with a value range of -1 to 1. In other words, the neuron input produced by both types of normalization will be passed on to the next layer according to that range. Minmax performance that exceeds z-score performance shows that the value that is highly influential on the BiLSTM architecture in this study is 0 to 1.

The prediction results produced by the Bi-LSTM model with minmax normalization have pretty good accuracy because they are close to the actual value. The model can predict according to the trend of energy use data properly due to the data ratio of 100:100. Precise prediction of energy demand can be utilized in planning infrastructure capacity, including power plants, distribution networks, and energy storage systems. Thus, wiser investments can be made and reduce potential imbalances between energy supply and demand.

5 Conclusion

This study used the Bidirectional LSTM (BiLSTM) deep learning method to analyze energy forecasting data. Tests using min-max normalization resulted in optimal performance with the best MAPE value of 7.7256%, RMSE of 0.1234%, and R2 of 0.6151%. The results show that Bi-LSTM's performance is effective in analyzing energy forecasting data using min-max normalization. Testing using z-score normalization in this study resulted in a slightly lower evaluation. The best MAPE produced is 10.5525%, with RMSE of 0.7627% and R2 of 0.4186%. In general, the use of Min-Max normalization results in better performance than Z-Score in analyzing energy forecasting.

A strategy that can be considered for future research is to apply other hyperparameter tunings, such as PSO. Using other tunings will generate new parameters for the deep learning model used. In addition, the use of different neurons at each layer of the deep learning model can also be applied to future research.

References

1. K. K. Jaiswal, C. R. Chowdhury, D. Yadav, R. Verma, S. Dutta, K. S. Jaiswal, SangmeshB, and K. S. K. Karuppasamy, Energy Nexus **7**, 100118 (2022)
2. E. E. Avalos, M. A. R. Licea, H. R. Gonzalez, A. E. Calderon, A. I. B. Gutierrez, and F. J. P. Pinal, 2020 IEEE Int. Autumn Meet. Power, Electron. Comput. ROPEC 2020 1 (2020)
3. W. Z. Khan, M. H. Rehman, H. M. Zangoti, M. K. Afzal, N. Armi, and K. Salah, Comput. Electr. Eng. **81**, 106522 (2020)
4. N. T. Mbungu, R. M. Naidoo, R. C. Bansal, M. W. Siti, and D. H. Tungadio, J. Energy Storage **29**, 101385 (2020)
5. T.-Z. Ang, M. Salem, M. Kamarol, H. S. Das, M. A. Nazari, and N. Prabaharan, Energy Strateg. Rev. **43**, 100939 (2022)
6. A. P. Wibawa, A. B. P. Utama, A. K. G. Akbari, A. F. Fadhilla, A. P. P. Triono, A. K. I. Paramarta, F. U. Setyaputri, and L. Hernandez, Knowl. Eng. Data Sci. **6**, 170 (2023)
7. A. Pranolo, Y. Mao, A. P. Wibawa, A. B. P. Utama, and F. A. Dwiyanto, Knowl. Eng. Data Sci. **5**, 53 (2022)
8. I. K. M. Jais, A. R. Ismail, and S. Q. Nisa, Knowl. Eng. Data Sci. **2**, 41 (2019)
9. M. Akhtar and S. Moridpour, J. Adv. Transp. **2021**, 1 (2021)
10. F. M. D. Mandagi, F. J. Paat, D. Tooy, S. E. Pakasi, and S. Wantasen, Bull. Soc. Informatics Theory Appl. **6**, 158 (2022)
11. A. Pranolo, Y. Mao, A. P. Wibawa, A. B. P. Utama, and F. A. Dwiyanto, IEEE Access **10**, 78423 (2022)
12. W. Zheng and G. Chen, IEEE Trans. Cybern. **52**, 13902 (2022)
13. S. Du, T. Li, Y. Yang, and S.-J. Horng, Neurocomputing **388**, 269 (2020)
14. Y. Zhang, R. Xiong, H. He, and M. G. Pecht, IEEE Trans. Veh. Technol. **67**, 5695 (2018)
15. A. Vrisna, H. Ar, M. Yasser, and S. Nazir, Knowl. Eng. Data Sci. **6**, 103 (2023)
16. S. Urolagin, N. Sharma, and T. K. Datta, Energy **231**, 120963 (2021)
17. I. M. Pires, F. Hussain, N. M. M. Garcia, P. Lameski, and E. Zdravevski, Futur. Internet **12**, 194 (2020)
18. I. Izonin, B. Ilchyshyn, R. Tkachenko, M. Greguš, N. Shakhovska, and C. Strauss, in *2022 12th Int. Conf. Adv. Comput. Inf. Technol.* (IEEE, 2022), pp. 480–484
19. D. Singh and B. Singh, Appl. Soft Comput. **97**, 105524 (2020)
20. J. Sun, X. Cao, H. Liang, W. Huang, Z. Chen, and Z. Li, AAAI 2020 - 34th AAAI Conf. Artif. Intell. 5875 (2020)
21. Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, and F. Ahmad, Trans. Emerg. Telecommun. Technol. **32**, (2021)
22. B. Lim and S. Zohren, Philos. Trans. R. Soc. A Math. Phys. Eng. Sci. **379**, (2021)
23. F. Masri, D. Saepudin, and D. Adytia, E-Proceeding Eng. **7**, 8544 (2020)
24. I. Verma, R. Ahuja, H. Meisheri, and L. Dey, Proc. - 2018 IEEE/WIC/ACM Int. Conf. Web Intell. WI 2018 651 (2019)
25. M. Yang and J. Wang, Procedia Comput. Sci. **199**, 18 (2022)
26. K. Wu, J. Wu, L. Feng, B. Yang, R. Liang, S. Yang, and R. Zhao, Int. Trans. Electr. Energy Syst. **31**, 1 (2021)
27. J. Santoso, E. I. Setiawan, C. N. Purwanto, and F. Kurniawan, Knowl. Eng. Data Sci. **4**, 38

(2021)
28. X. Li, Z. Li, X. Zheng, X. Yang, and X. Yu, Electronics **12**, 327 (2023)
29. A. K. Rotty, T. S. Dewayana, and A. N. Habyba, (2022)
30. M. A. Hasanah, S. Soim, and A. S. Handayani, J. Appl. Informatics Comput. **5**, 103 (2021)
31. L. Linde, D. Sjödin, V. Parida, and H. Gebauer, Res. Manag. **64**, 43 (2021)
32. M. M. Archibald, R. C. Ambagtsheer, M. G. Casey, and M. Lawless, Int. J. Qual. Methods **18**, 160940691987459 (2019)
33. F. Martinez-Plumed, L. Contreras-Ochando, C. Ferri, J. Hernandez-Orallo, M. Kull, N. Lachiche, M. J. Ramirez-Quintana, and P. Flach, IEEE Trans. Knowl. Data Eng. **33**, 3048 (2021)
34. A. P. Joshi and B. V. Patel, Orient. J. Comput. Sci. Technol. **13**, 78 (2021)
35. T. Darmawan, Int. J. Emerg. Trends Eng. Res. **8**, 2375 (2020)
36. D. Chicco, M. J. Warrens, and G. Jurman, PeerJ Comput. Sci. **7**, 1 (2021)
37. A. B. Al-Ghamdi, S. Kamel, and M. Khayyat, Proc. - 2021 IEEE 4th Natl. Comput. Coll. Conf. NCCC 2021 (2021)
38. H. W. Herwanto, A. N. Handayani, A. P. Wibawa, K. L. Chandrika, and K. Arai, in *2021 7th Int. Conf. Electr. Electron. Inf. Eng.* (IEEE, 2021), pp. 1–3