

# Journal Unique Visitors Forecasting Based on Multivariate Attributes Using CNN

Aderyan Reynaldi Fahrezza Dewandra<sup>a,1</sup>, Aji Prasetya Wibawa<sup>a,2,\*</sup>, Utomo Pujianto<sup>a,3</sup>, Agung Bella Putra Utama<sup>a,4</sup>, Andrew Nafalski<sup>b,5</sup>

<sup>a</sup> Universitas Negeri Malang, Jl. Semarang no. 5, Malang, 65145, Indonesia

<sup>b</sup> University of South Australia, 101 Currie St, Adelaide SA 5001, Australia

<sup>1</sup> aderey2599@gmail.com; <sup>2</sup> aji.prasetya.ft@um.ac.id\*; <sup>3</sup> utomo.pujianto.ft@um.ac.id; <sup>4</sup> agungbpu02@gmail.com; <sup>5</sup> andrew.nafalski@unisa.edu.au

\* corresponding author

---

## ARTICLE INFO

### Article history:

Received 15 Feb 2022

Revised 20 Apr 2022

Accepted 23 May 2022

### Keywords:

Forecasting

Multivariate

Unique Visitors

1D Convolutional Neural Network

## ABSTRACT

Forecasting is needed in various problems, one of which is forecasting electronic journal unique visitors. Although forecasting cannot produce accurate predictions, using the proper method can reduce forecasting errors. In this research, forecasting is done using the Deep Learning method, which is often used to process two-dimensional data, namely convolutional neural network (CNN). One-dimensional CNN comes with 1D feature extraction suitable for forecasting 1D time-series problems. This study aims to determine the best architecture and increase the number of hidden layers and neurons on CNN forecasting results. CNN performance was measured using the root mean squared error (RMSE) in various architectural scenarios. Based on the study results, the best results were obtained with an RMSE value of 2.314 using an architecture of 2 hidden layers and 64 neurons in Model 1. Meanwhile, the significant effect of increasing the number of hidden layers on the RMSE value was only found in Model 1 using 64 or 256 neurons.

Copyright © 2017 International Journal of Artificial Intelligence Research.

All rights reserved.

## I. Introduction

Forecasting is an activity to predict conditions that will occur in the future by utilizing all conditions that occurred in the past [1]. Forecasting can help solve various problems that require a picture of future conditions. Forecasting can also determine the right decision based on considerations of circumstances that have occurred previously [2]. Forecasting does not produce definite predictions because there is always a problem of uncertainty in the future [3]. For this reason, the proper selection of forecasting methods can reduce the error rate to produce an optimal prediction. Currently, various problems require forecasting to determine strategies or decisions based on predictions of future conditions, one of which is forecasting unique visitors (sessions) in electronic journals.

Research on session forecasting has been done previously. Session forecasting using the multilayer perceptron (MLP) method produces an RMSE value of 0.137826 [4]. This method is also combined with the single exponential smoothing method in session forecasting, which produces an RMSE value of 0.7554 [5]. Another session forecasting uses the long short-term memory (LSTM) method and produces an RMSE value of 13.76 [6]. The LSTM smoothing variant has also been applied to session forecasting, producing a MAPE value of 0.08098 [7]. In addition, the backpropagation neural network (BPNN) method is also used to predict the session and generate a MAPE of 0.301 [8]. Based on the value of the results obtained, forecasting in previous studies proved effective in solving prediction problems. However, they only use a single attribute, namely sessions, so the time-series analysis is univariate. While the dataset used has four attributes: sessions, page views, visitors, and new visitors.

Therefore, this study focuses on multivariate analysis of e-journal sessions using a convolutional neural network (CNN). CNN is widely used because it has good performance in various fields that use two-dimensional data, such as computer vision and image processing [9]. However, recently the

one-dimensional CNN (1D CNN) model has begun to be applied to prediction tasks involving time series with satisfactory results [10]. 1D CNN can predict time series problems by extracting one-dimensional features from the data used [11], [12]. A CNN-based multi-input model was employed in research [13] to estimate the power generated by a wave energy converter (WEC) device. Four-variable time-series input data is transformed into two-dimensional data. The findings indicated that RMSE of multi-input CNN (3.11) outperformed other supervised modeling networks such as Artificial Neural Network (2144.83), Support Vector Machine (34.88), Robust Linear Regression (35.15), Medium Tree (23.36), and Boosted Tree (20.83). 1D CNN was used in the study [14] to forecast monthly rainfall for chosen regions in eastern Australia. Climate indices are the different atmospheric factors that cause rain to fall. Several meteorological features and climate indicators are gathered from various sources to be utilized as rainfall predictors for certain places at specific year periods. The study results show that the RMSE of CNN (142.133) outperforms the forecasting model of the Australian Community Climate and Earth-System Simulator (ACCESS) meteorological bureau (179.139) and conventional multi-layered perceptron (158.074). CNN 1D has also been used in research [15] to predict stock price movements in the Chinese stock market using stock data. Stock data is a one-dimensional time series data set with five characteristics: the opening price, high price, low price, closing price, and stock volume. According to the study's findings, a deep learning approach based on Convolutional Neural Networks may be utilized to forecast the movement of Chinese stock values.

This study aims to use 1D CNN for forecasting multivariate time-series data. All attributes in the dataset will be used to generate unique visitor predictions. This study also aims to determine the effect of increasing the number of hidden layers and neurons on the 1D CNN method results. 1D CNN performance will be measured using the root mean squared error (RMSE) method for all different architectures.

## II. Method

In this study, the journal sessions forecasting process used the CNN method. The forecasting process has several stages of the research flow, as shown in Figure 1.



Fig. 1. Research Flow

Based on Figure 1, the activity data of the journal portal visitors is collected first from the e-journal site. After the dataset is obtained, the following process is data normalization. The CNN method is used in the forecasting process by applying several different architectures. In the final stage, CNN's forecasting performance is evaluated using the RMSE method. A complete explanation will be given in the following sub-chapters.

### A. Data Collection

The dataset in this study is the visitor activity data of the Knowledge Engineering and Data Science (KEDS) journal portal. The dataset used starts from January 1, 2017, to December 31, 2020. The dataset has four attributes: sessions, page views, visitors, and new visitors. The dataset is used to predict session variables. The session was chosen as the target of forecasting because it is an indicator of the success of an electronic journal [6]. The increasing number of sessions per day illustrates the wide distribution and high public interest in electronic journals, accelerating the journal accreditation system [7]. The sample dataset visualization for 2020 can be seen in Figure 2.

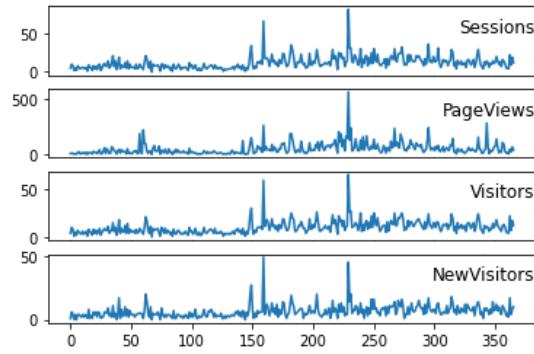


Fig. 2.2020 Dataset

This study uses three models based on training and testing data used. The distribution of the dataset is shown in Table 1.

Table 1. Data Usage of Each Model

Model	Data Training	Data Testing
Model 1	2018	2019
Model 2	2019	2020
Model 3	2018 – 2019	2020

### B. Data Normalization

The normalization method used is min-max normalization. Normalization is important because the data on each variable has a different range [16]. This method can change the range of each variable to a range between 0 and 1 [17], [18]. Min-max normalization is performed using (1) [19].

$$x^* = \frac{x - \min}{\max - \min} \quad (1)$$

Where  $x^*$  is the normalized value,  $x$  is the original value,  $\min$  is the lowest value from the dataset, and  $\max$  is the highest value in the dataset.

### C. Forecasting Process Using 1D Convolutional Neural Network (CNN)

1D CNN consists of a 1D input layer, 1D convolution layer, 1D pooling layer, fully-connected layer, and output layer [20], as shown in Figure 3. The input layer helps receive one-dimensional data. After inputting the data, the data is passed to the convolutional layer. In the convolutional layer, there is a 1D filter that is used to convolute the data [21]. Convolution is used to extract features from input data. The 1D filter shifts along the input data to perform dot product operations with the appropriate receptive field, and then the results are entered into the input feature map [22]. The stride value determines the magnitude of the filter shift. The lower the stride value, the more detailed the feature information will be and the more and longer the calculations will take. The resulting input feature map is forwarded to the activation function to generate a neuron output feature map from convoluted features [23].

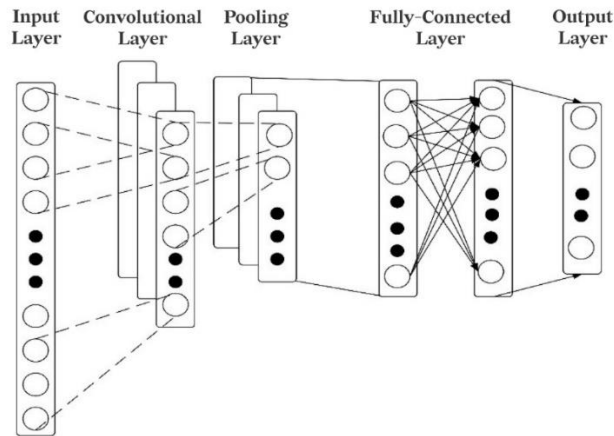


Fig. 3. The general structure of 1D CNN

The pooling layer is tasked with reducing the dimensions of the feature map output from the convolutional layer to reduce the number of parameters and control overfitting [24]. This layer also aims to reduce the computational complexity of the network [25]. The most common types of pooling are average pooling and max pooling. Average pooling is used to extract a single average value from a group of neurons. While max-pooling extracts a single maximum value from a group of neurons, as shown in Figure 4.

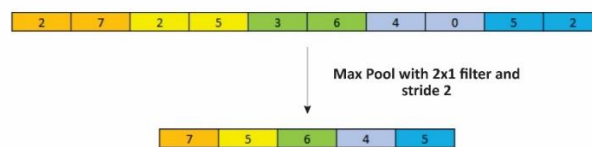


Fig. 4. Max Pool Illustration

The fully-connected layer is helpful for processing data using MLP to get the expected results [26]. This layer consists of many neurons fully connected with other layers [27]. This layer also gets input from the output of the feature extraction layer (convolution layer and pooling layer) [28]. Through this layer, the input is changed until the final result is obtained [29]. The output of the fully-connected layer is passed to the output layer. The output layer is responsible for displaying predictions [30].

In this study, the application of the 1D CNN method uses architecture, as shown in Table 2. This architectural variation was adopted from studies [31]–[33] with minor changes based on need. This architecture is used in every forecasting model that has been designed.

Table 2. 1D CNN Architecture

Layer	Type	Number of Layers	Parameter							
			No. of Neurons	No. of Filters	Kernel size	Activation function	Pool size	Dropout rate	Loss function	Optimizer
Conv1D	Convolutional Layer	1	-	64	1	ReLU	-	-	-	-
MaxPool1D	Pooling Layer	1	-	-	-	-	1	0.5	-	-
Flatten	Flatten Layer	1	-	-	-	-	-	-	-	-
Dense	Fully Connected Layer	{ 2, 3, 4, 5, 6, 7, 8, 9, 10 }	{ 16, 32, 64, 128, 256 }	-	-	ReLU	-	0.2	MSE	Adam
Dense	Output Layer	1	1	-	-	Linear	-	-	-	-

Based on Table 2, the convolution layer uses a one-dimensional convolution layer with 64 filters, 1 kernel size, and the ReLU activation function. The pooling layer uses a one-dimensional max pooling layer with a pool of size 1, and the dropout rate is 0.5. The flatten layer is used to convert the

output matrix of the previous layer into a one-dimensional vector. The different hidden layers and neurons are used in a fully connected layer. The hidden layer starts from 2 to 10, while the number of neurons uses 16, 32, 64, 128, and 256. This layer uses the ReLU activation function, the dropout rate is 0.2, the type of loss function is mean squared error (MSE), and the optimizer uses adam. Here, the batch size is 16, and the maximum epoch is 500.

#### D. Evaluation

The performance of the forecasting method was measured using the root mean squared error (RMSE) method. RMSE works by calculating the error value to detect irregularities or outliers in the designed forecasting system [34]. The smaller the RMSE value, the better the performance of the forecasting model. Equation (2) shows the RMSE formulae [35].

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y'_i - y_i)^2} \quad (2)$$

Where  $y'_i$  is the predicted value,  $y_i$  is the actual value, and  $n$  is the number of data.

### III. Result and Discussion

A total of 135 configurations were carried out. Each configuration has ten evaluated experiments. Table 3 shows the average results of RMSE in every hidden layer scenario and model.

Table 3. RMSE of Forecasting Results

Hidden layer	Neuron														
	Model 1					Model 2					Model 3				
	16	32	64	128	256	16	32	64	128	256	16	32	64	128	256
2	3.166	2.885	2.314	3.233	2.586	3.354	3.631	3.605	3.302	3.186	2.934	3.140	3.079	2.747	2.331
3	4.060	3.144	3.580	2.940	2.688	3.789	4.044	3.578	3.473	3.449	3.669	3.576	3.251	3.317	3.225
4	5.056	4.387	3.005	3.633	3.882	5.297	5.187	4.747	4.261	4.582	5.188	3.986	4.165	5.341	3.954
5	5.346	4.456	4.424	3.412	6.908	5.112	5.010	4.482	4.122	4.897	5.573	4.843	4.738	3.864	3.990
6	6.317	5.355	6.019	4.894	7.548	5.539	5.615	5.293	5.634	5.319	6.689	5.553	5.692	5.610	5.767
7	6.936	7.044	5.054	6.110	9.697	6.585	6.384	5.931	5.542	5.640	7.304	6.524	5.700	6.761	8.813
8	6.993	6.901	5.882	7.634	9.038	7.372	7.022	6.129	7.530	8.217	7.376	7.483	7.053	8.596	9.311
9	7.977	6.905	7.187	9.244	9.549	7.833	8.146	7.938	8.981	9.279	7.239	6.225	6.888	8.161	9.475
10	8.304	7.330	7.783	8.557	9.779	9.010	8.075	8.556	8.905	9.272	7.988	7.469	8.396	8.872	9.435

Based on the forecasting results in Table 3, Model 1 has the best RMSE of 2.314 using 2 hidden layers and 64 neurons. The use of different architectures makes the RMSE is gradually decreased. Hidden layers increase the RMSE, especially when using 16 neurons. Meanwhile, the increased number of neurons used relatively reduces the RMSE.

In Model 2, the best RMSE is 3.186, using 2 hidden layers and 256 neurons. The increase in the hidden layer tends to increase the RMSE. On the other hand, the increase of neurons tends to cause a decrease in the RMSE. In other words, the increase of neurons and hidden layers has a different effect on RMSE.

While the best RMSE results in Model 3 are 2.331 using an architecture with 2 hidden layers and 256 neurons. As in Model 1, the increase of hidden layers also increases the RMSE. In addition, the greater the number of neurons used, the RMSE value decreases relatively. Based on comparing the RMSE of the three models in Table 2, the best CNN architecture for multivariate analysis of journal unique visitors is 2 hidden layers with 64 and 256 neurons.

A significance test was carried out using the Paired Sample T-Test to determine the effect of increasing the number of hidden layers and neurons used on the RMSE. A significance value of less than 0.05 indicates that the paired sample changes significantly. The result shows a significant effect on the difference in treatment for each variable. On the other hand, a significance value of more than 0.05 indicates no significant change or effect of the difference in treatment on each variable. The test

results of the significance of increasing the number of hidden layers and neurons on the resulting RMSE can be seen in Table 4 and Table 5.

Table 4. The Result of Paired T-test to Test the Significant of Increasing Hidden Layers in Every Neuron Scenario

Model	Neuron				
	16	32	64	128	256
Model 1	0.9618	0.1492	<b>0.0294</b>	0.1435	<b>0.0393</b>
Model 2	0.9705	0.8074	0.3089	0.407	0.9478
Model 3	0.9913	0.218	0.1361	0.7894	0.409

Table 5. Significance Test Result for the Increase of the Neurons in Each Hidden Layer Scenario

Model	Hidden layer								
	2	3	4	5	6	7	8	9	10
Model 1	0.0921	0.0939	0.0952	0.0951	0.0982	0.0991	0.1005	0.1029	0.1037
Model 2	0.0935	0.0942	0.0973	0.0966	0.0984	0.1005	0.1016	0.1045	0.1062
Model 3	0.0926	0.0935	0.0963	0.0974	0.0996	0.1001	0.1025	0.1008	0.1043

Based on the results in Table 4 and Table 5, it can be proven that an increase in the number of hidden layers only has a significant effect on the RMSE value in Model 1 with the use of 64 and 256 neurons. At the same time, the increase in the number of neurons has no significant effect on the RMSE value in all existing models.

#### IV. Conclusion

From the results of this study, it can be concluded that the use of CNN for Journal Unique Visitors Forecasting Based on Multivariate Attributes is visible. The addition of the number of hidden layers of neurons in the CNN method in this study has different effects on each model. The resulting RMSE does not produce a general pattern due to the increase of hidden layers of neurons. Based on the significance test result, a significant effect of adding the number of hidden layers to the RMSE is only found in Model 1 with 64 and 256 neurons. The future research will propose using hyperparameter tuning to find the best model for CNN-based multivariate time-series forecasting.

#### References

- [1] Z. Hao, V. P. Singh, and Y. Xia, "Seasonal Drought Prediction: Advances, Challenges, and Future Prospects," *Rev. Geophys.*, vol. 56, no. 1, pp. 108–141, Mar. 2018, doi: 10.1002/2016RG000549.
- [2] R. Majid, "Advances in Statistical Forecasting Methods: An Overview," *Econ. Aff.*, vol. 63, no. 4, 2018, doi: 10.30954/0424-2513.4.2018.5.
- [3] F. Petropoulos and S. Makridakis, "Forecasting the novel coronavirus COVID-19," *PLoS One*, vol. 15, no. 3, pp. 1–8, 2020, doi: 10.1371/journal.pone.0231236.
- [4] A. P. Wibawa, W. Lestari, A. B. P. Utama, I. T. Saputra, and Z. N. Izdihar, "Multilayer Perceptron untuk Prediksi Sessions pada Sebuah Website Journal Elektronik," *Indones. J. Data Sci.*, vol. 1, no. 3, Dec. 2020, doi: 10.33096/ijodas.v1i3.15.
- [5] M. A. B. Ferdinand, A. P. Wibawa, I. A. E. Zaeni, and H. A. Rosyid, "Single Exponential Smoothing-Multilayer Perceptron Untuk Peramalan Pengunjung Unik," *J. Mob. Forensics*, vol. 2, no. 2, pp. 62–70, 2020, doi: https://doi.org/10.12928/mf.v2i2.2034.
- [6] A. P. Wibawa, I. T. Saputra, A. B. P. Utama, W. Lestari, and Z. N. Izdihar, "Long Short-Term Memory to Predict Unique Visitors of an Electronic Journal," in *2020 6th International Conference on Science in Information Technology (ICSITech)*, Oct. 2020, pp. 176–179, doi: 10.1109/ICSITech49800.2020.9392031.
- [7] A. P. Wibawa, R. R. Ula, A. B. P. Utama, M. Y. Chuttur, A. Pranolo, and Haviluddin, "Forecasting e-Journal Unique Visitors using Smoothed Long Short-Term Memory," in *2021 7th International*

- Conference on Electrical, Electronics and Information Engineering (ICEEIE)*, Oct. 2021, pp. 609–613, doi: 10.1109/ICEEIE52663.2021.9616628.
- [8] A. P. Wibawa, Z. N. Izdihar, A. B. P. Utama, L. Hernandez, and Haviluddin, “Min-Max Backpropagation Neural Network to Forecast e-Journal Visitors,” in *2021 International Conference on Artificial Intelligence in Information and Communication (ICAIIIC)*, Apr. 2021, pp. 052–058, doi: 10.1109/ICAIIIC51459.2021.9415197.
- [9] S. Khan, H. Rahmani, S. A. A. Shah, and M. Bennamoun, “A Guide to Convolutional Neural Networks for Computer Vision,” *Synth. Lect. Comput. Vis.*, vol. 8, no. 1, pp. 1–207, Feb. 2018, doi: 10.2200/s00822ed1v01y201712cov015.
- [10] S. Cavalli and M. Amoretti, “CNN-based multivariate data analysis for bitcoin trend prediction,” *Appl. Soft Comput.*, vol. 101, p. 107065, 2021, doi: 10.1016/j.asoc.2020.107065.
- [11] G. T. Taye, H. J. Hwang, and K. M. Lim, “Application of a convolutional neural network for predicting the occurrence of ventricular tachyarrhythmia using heart rate variability features,” *Sci. Rep.*, vol. 10, no. 1, pp. 1–7, 2020, doi: 10.1038/s41598-020-63566-8.
- [12] Z. Zhang, J. Tian, W. Huang, L. Yin, W. Zheng, and S. Liu, “A Haze Prediction Method Based on One-Dimensional Convolutional Neural Network,” *Atmosphere (Basel)*, vol. 12, no. 10, pp. 1–12, 2021, doi: 10.3390/atmos12101327.
- [13] C. Ni, X. Ma, and Y. Bai, “Convolutional neural network based power generation prediction of wave energy converter,” *ICAC 2018 - 2018 24th IEEE Int. Conf. Autom. Comput. Improv. Product. through Autom. Comput.*, no. September, pp. 1–6, 2018, doi: 10.23919/ICAC.2018.8749043.
- [14] A. Haidar and B. Verma, “Monthly Rainfall Forecasting Using One-Dimensional Deep Convolutional Neural Network,” *IEEE Access*, vol. 6, pp. 69053–69063, 2018, doi: 10.1109/ACCESS.2018.2880044.
- [15] S. Chen and H. He, “Stock Prediction Using Convolutional Neural Network,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 435, no. 1, 2018, doi: 10.1088/1757-899X/435/1/012026.
- [16] R. Fukuoka, H. Suzuki, T. Kitajima, A. Kuwahara, and T. Yasuno, “Wind Speed Prediction Model Using LSTM and 1D-CNN,” *J. Signal Process.*, vol. 22, no. 4, pp. 207–210, 2018, doi: 10.2299/jsp.22.207.
- [17] A. Gumaei, M. M. Hassan, M. R. Hassan, A. Alelaiwi, and G. Fortino, “A Hybrid Feature Extraction Method with Regularized Extreme Learning Machine for Brain Tumor Classification,” *IEEE Access*, vol. 7, pp. 36266–36273, 2019, doi: 10.1109/ACCESS.2019.2904145.
- [18] S. Grover, S. Bhartia, Akshama, A. Yadav, and K. R. Seeja, “Predicting Severity of Parkinson’s Disease Using Deep Learning,” *Procedia Comput. Sci.*, vol. 132, no. Iccids, pp. 1788–1794, 2018, doi: 10.1016/j.procs.2018.05.154.
- [19] Y. Ju, G. Sun, Q. Chen, M. Zhang, H. Zhu, and M. U. Rehman, “A Model Combining Convolutional Neural Network and LightGBM Algorithm for Ultra-Short-Term Wind Power Forecasting,” *IEEE Access*, vol. 7, pp. 28309–28318, 2019, doi: 10.1109/ACCESS.2019.2901920.
- [20] L. Ni, Y. Li, X. Wang, J. Zhang, J. Yu, and C. Qi, “Forecasting of Forex Time Series Data Based on Deep Learning,” *Procedia Comput. Sci.*, vol. 147, pp. 647–652, 2019, doi: 10.1016/j.procs.2019.01.189.
- [21] S. Malek, F. Melgani, and Y. Bazi, “One-dimensional convolutional neural networks for spectroscopic signal regression,” *J. Chemom.*, vol. 32, no. 5, pp. 1–17, 2018, doi: 10.1002/cem.2977.
- [22] W. Ng *et al.*, “Convolutional neural network for simultaneous prediction of several soil properties using visible/near-infrared, mid-infrared, and their combined spectra,” *Geoderma*, vol. 352, no. January, pp. 251–267, 2019, doi: 10.1016/j.geoderma.2019.06.016.
- [23] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman, “1D convolutional neural networks and applications: A survey,” *Mech. Syst. Signal Process.*, vol. 151, p. 107398, 2021, doi: 10.1016/j.ymsp.2020.107398.
- [24] R. B. Arif, M. A. B. Siddique, M. M. R. Khan, and M. R. Oishe, “Study and observation of the variations of accuracies for handwritten digits recognition with various hidden layers and epochs using

- convolutional neural network,” *4th Int. Conf. Electr. Eng. Inf. Commun. Technol. iCEEiCT 2018*, pp. 112–117, 2018, doi: 10.1109/CEEICT.2018.8628078.
- [25] S. Yi, J. Ju, M.-K. Yoon, and J. Choi, “Grouped Convolutional Neural Networks for Multivariate Time Series,” Mar. 2017.
- [26] M. A. Syarifudin *et al.*, “Hotspot Prediction Using 1D Convolutional Neural Network,” *Procedia Comput. Sci.*, vol. 179, no. 2019, pp. 845–853, 2021, doi: 10.1016/j.procs.2021.01.073.
- [27] P. Sharma, S. Chandan, and B. P. Agrawal, “Vibration Signal-based Diagnosis of Defect Embedded in Outer Race of Ball Bearing using 1-D CNN,” *2020 Int. Conf. Comput. Perform. Eval. ComPE 2020*, no. July, pp. 531–536, 2020, doi: 10.1109/ComPE49325.2020.9199994.
- [28] A. Mulyadi and E. C. Djamal, “Sunshine Duration Prediction Using 1D Convolutional Neural Networks,” *Proc. 2019 6th Int. Conf. Instrumentation, Control. Autom. ICA 2019*, no. August, pp. 77–81, 2019, doi: 10.1109/ICA.2019.8916751.
- [29] E. Hoseinzade and S. Haratizadeh, “CNNpred: CNN-based stock market prediction using a diverse set of variables,” *Expert Syst. Appl.*, vol. 129, pp. 273–285, 2019, doi: 10.1016/j.eswa.2019.03.029.
- [30] M. I. Khan and R. Maity, “Hybrid Deep Learning Approach for Multi-Step-Ahead Daily Rainfall Prediction Using GCM Simulations,” *IEEE Access*, vol. 8, pp. 52774–52784, 2020, doi: 10.1109/ACCESS.2020.2980977.
- [31] A. Parashar and A. Sonker, “Application of hyperparameter optimized deep learning neural network for classification of air quality data,” *Int. J. Sci. Technol. Res.*, vol. 8, no. 11, pp. 1435–1443, 2019.
- [32] Y. Li, Z. Zhu, D. Kong, H. Han, and Y. Zhao, “EA-LSTM: Evolutionary attention-based LSTM for time series prediction,” *Knowledge-Based Syst.*, vol. 181, p. 104785, Oct. 2019, doi: 10.1016/j.knosys.2019.05.028.
- [33] M. Tovar, M. Robles, and F. Rashid, “PV Power Prediction, Using CNN-LSTM Hybrid Neural Network Model. Case of Study: Temixco-Morelos, México,” *Energies*, vol. 13, no. 24, p. 6512, 2020, doi: 10.3390/en13246512.
- [34] A. Jadidi, R. Menezes, N. de Souza, and A. de Castro Lima, “A Hybrid GA–MLPNN Model for One-Hour-Ahead Forecasting of the Global Horizontal Irradiance in Elizabeth City, North Carolina,” *Energies*, vol. 11, no. 10, p. 2641, Oct. 2018, doi: 10.3390/en11102641.
- [35] X. Qiu, L. Zhang, P. Nagarathnam Suganthan, and G. A. J. Amaratunga, “Oblique Random Forest Ensemble via Least Square Estimation for Time Series Forecasting,” *Inf. Sci. (Ny)*, vol. 420, pp. 249–262, 2017, doi: 10.1016/j.ins.2017.08.060.