A Hybrid Neural Network-Time Series Regression Model for Intermittent Demand Forecasting Data

Amri Muhaimin[*1)], Aviolla Terza Damaliana[2)], Muhammad Nasrudin[3)], Prismahardi Aji Riyantoko[4)], Nabilah Selayanti[5)], Shafira Amanda Putri[6)]

[1, 2, 3, 5, 6)]Faculty of Computer Science, UPN Veteran Jawa Timur, Surabaya, Indonesia
[4)]Department of Information and Communication Systems, Okayama University, Okayama, Japan
[1*]amri.muhaimin.stat@upnjatim.ac.id, [2]aviolla.terza.sada@upnjatim.ac.id, [3]nasrudin.fasilkom@upnjatim.ac.id, [4]pnai2m3s@s.okayama-u.ac.jp, [5]22083010013@student.upnjatim.ac.id, [6]2208301008@student.upnjatim.ac.id

ARTICLE INFO

Article history:
Received 24 June 2025
Revised 4 November 2025
Accepted 20 November 2025
Available online 25 November 2025

Keywords:
Decision-making
Exponential Smoothing
Forecasting
Hybrid Model
Intermittent Data

ABSTRACT

Forecasting is a vital tool that helps us make informed decisions by predicting future events based on past data. For forecasts to be accurate, it is important that the data is reliable, complete, and consistent. Yet, the intermittent data is a unique data that is challenging to forecast. Intermittent data contains a characteristic that the data has a lot of long zeros in some periods. The zero value will influence the model to generate a forecasting model. This study aims to tackle those problems by applying a hybrid approach. We integrate the regression model and neural network to create a novel approach for forecasting intermittent data. The dataset used for this data is from Kaggle, sales at Walmart supermarket for one category only. The sales data always produce an intermittent demand pattern, because not every day are the items always sold to customers. This irregular pattern makes the data difficult to forecast using a naïve approach, such as the Croston method, exponential smoothing, and ARIMA. To evaluate the performance of our model, some metrics were calculated. We use mean squared error, root mean squared error, and root mean squared scaled error. The result shows that our proposed method outperforms the benchmark model, with an RMSSE of 0.98, which is the lowest compared to other benchmark models in the root mean squared scaled error value. This result shows promise as an exciting solution for overcoming the challenges posed by irregular data in future forecasting tasks.

1. Introduction

Time series analysis is a popular technique used to make predictions about future events. However, some data sets can be quite challenging to predict, especially when they contain a lot of zero values. A common example of this is sales data, such as the sale of vehicles and a lot of products that are difficult to sell. These sales don't happen every day, and periods without any sales are recorded as zeros in the data [1]. Traditional time series methods like Autoregressive Integrated Moving Average (ARIMA) struggle with this type of data because the large number of zeros can lead to inaccurate predictions, sometimes even generating negative forecast values. This type of data, often referred to as intermittent data, has its own unique characteristics. Intermittent data can be categorized into four types: intermittent, lumpy, erratic, and smooth [2]. Among these, the intermittent and lumpy types contain more zero values compared to erratic and smooth data [3] [4]. Given the challenges of predicting intermittent sales data, it's clear that a more specialized approach is required to address this issue effectively [5].

This study aims to tackle the challenges of forecasting data with many zero values by using a hybrid model that combines Neural Networks and Time Series Regression. This approach is particularly well-suited for datasets like these, as it helps produce more accurate and realistic predictions [6]. To assess how well the model performs, we also use the Root Mean Square Scaled Error (RMSSE), which is ideal for datasets that include a significant number of zeros. Unlike other metrics, such as Mean Absolute Error (MAE) or Mean Absolute Percentage Error (MAPE), which can lead to unreliable results when there are many zeros in the data, the RMSSE offers a more stable and accurate measure of model performance [7].

Research on intermittent is rarely conducted, yet it is common. Various models are developed, such as the Neural Network method, Croston, and ARIMA. Hence, the results were not good enough and sometimes tended to stagnate. Cause of the data has too many zero values [8]. This approach aims to increase the goodness of the resulting model. Combined with the Exponential Smoothing method, they managed to get the best results, but were not robust for all types of intermittent data. Then, Ref [9] used the Deep Learning approach directly with the Recurrent Neural Network model. This approach is quite expensive for modeling intermittent data. The results are not much different from the exponential smoothing or Croston methods. So, it is not efficient enough to be used on intermittent data. Furthermore, the Neural Network model uses a modified error criterion [10]. The error criterion is named sMDL, and it can optimize the calculation process of the NN method. But sometimes, there are still obstacles, such as the model becoming underfit or overfitting.

Ref [11] was the first researcher to introduce a mixed method to improve the accuracy of the forecasting model. A mixed method between Time Series Regression and Neural Networks can provide significant results [12] [13]. Using an activation function that does not produce negative values, Neural Networks can provide maximum results on the model [14]. This mixed concept relies on the Time Series Regression method as the main model. In the regression model, there is an error or residual value, which is then re-modeled using the Neural Network model. After that, the error forecast results are returned to the Regression model to produce forecast values. The forecast results are evaluated using the RMSSE value, which is very suitable for sparse data. The selection of parameters in the Neural Networks model uses optimization methods such as grid search [15]. That way, the forecast results are very robust, even with sparse data.

The data used in this study comes from daily sales records at a supermarket, and the period is five years, with a total of 1941 days of data. This dataset is part of the M5 competition hosted on the Kaggle website [16]. For the purposes of this study, the researcher selected just one specific dataset to focus on. The key variables used in the analysis include the running average of the sales data and its lag values, both of which are critical in understanding and predicting sales trends.

2. Methods

To use time-series regression, the model needs some input, in this case, a variable. To decide the variable to train the time-series regression model, the autocorrelation function analysis is performed. Choosing the lag as a variable based on the partial autocorrelation function plot, the time-series regression model can be done. After the time-series regression model is produced, the partial hypothesis testing is performed to select which lag is significant. After the error collected from the time-series regression model is moved into a neural network, it is used in the further modeling phase.

The time series model regression has similarities with a linear regression model. Assuming the output as a series, so we have Y_t with $t = 1,2,\dots,n$, and will be influenced by other variables in the input layer [17]. Then, the time series regression model is combined with a neural network. The neural network in this case will tackle the nonlinear problems in the data. So, the regression model will be robust in terms of forecasting the nonlinear part of the data [18]. The neural network equation with one hidden layer consisting of K inputs, one hidden layer consisting of j units, and connected to the output, can be written as follows:

$$\hat{Y}_t = f^0\left[b^0 \sum_{j=1}^{J}\left[w_j^0 f_j^1\left(b_j^1 + \sum_{i=1}^{K} w_{ji}^1 x_{i,t}\right)\right]\right] \tag{1}$$

 DOI: https://doi.org/10.52435/jaiit.v7i2.704

where f_j^1 is the activation function at the jth hidden layer and f^0 is the activation function at the output layer, w_j^0 is connecting the weight from the jth hidden layer to the output layer and w_{ji}^1 is connecting the weight from the i-th input to the j-th hidden layer [19].

This hybrid model combines two time series analysis methods: regression and neural networks. Thus, this hybrid modeling consists of two main stages. The first stage is the original data, which is modeled using time series regression and predictor variables in the form of significant PACF lags. The results of this regression will produce residuals that are the difference between the original and predicted data [20]. The second stage is to model the residuals of the regression model with a neural network using input in the form of significant PACF lags from the residuals of the regression model. The final predicted value of this hybrid model is the sum of the predicted results from the regression model and the neural network.

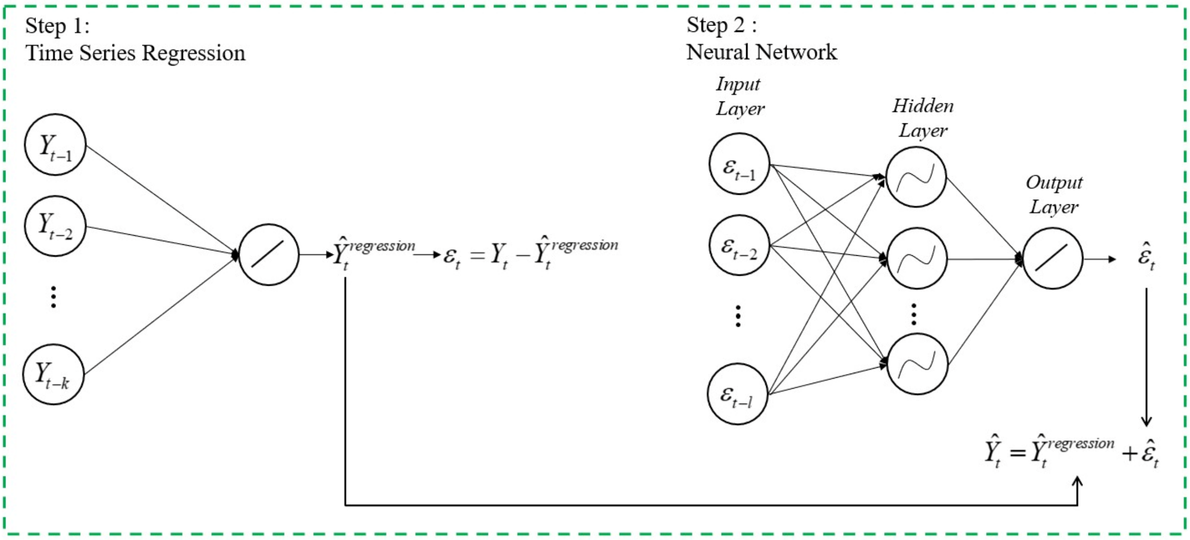

Figure 1. The Hybrid Method of Neural-Network and Time-Series Regression

This work contains some steps based on Figure 1:

a. Divide the original data into two parts: training data and testing data.
b. Model the original data using time series regression
   i. Identify significant PACF lags from the training data.
   ii. Model the training data using time series regression with predictor variables being lagged data identified in the previous step.
   iii. Calculate predicted values, residuals, and forecasts for the testing data based on the regression model.
c. Model the residuals of the regression model using a Neural Network:
   i. Identify significant PACF lags from the residuals.
   ii. Determine the number of neurons in the hidden layer using the cross-validation method.
   iii. Estimate the parameters of the Neural Network model using the backpropagation algorithm.
   iv. Calculate predictions and forecasts from the best Neural Network model.
d. Combine the predicted values from the regression model with the Neural Network as the predicted values for training data, and the forecast values from the regression model with the Neural Network as the predicted values for testing data.
e. Calculate the MSE, RMSE, and RMSSE values as the evaluation metric for the hybrid model.

Model evaluation in this study uses the calculation of the Root Mean Square Scaled Error (RMSSE) value [21]. In addition to using RMSSE, it also uses the Mean Squared Error (MSE) and Root Mean Squared Error (RMSE). The equations of MSE, RMSE, and RMSSE are given in Equations (2-4), where N is the length of the training data and L is the length of the testing data [22] [21].

$$MSE = \frac{\sum_{l=1}^{L}(y_l - \hat{y}_l)^2}{L} \tag{2}$$

$$RMSE = \sqrt{\frac{\sum_{l=1}^{L}(y_l - \hat{y}_l)^2}{L}} \tag{3}$$

$$RMSSE = \sqrt{\frac{\frac{1}{L}\sum_{l=N+1}^{N+L}(y_l - \hat{y}_l)^2}{\frac{1}{N-1}\sum_{n=2}^{N}(y_n - y_{n-1})^2}} \tag{4}$$

These metrics are all connected, they each provide different perspectives on model performance. MSE measures the average of the squared differences between predicted and actual values, giving more weight to larger errors. However, since the result is in squared units, it can be harder to interpret. To make it more understandable, RMSE takes the square root of MSE, bringing the error measurement back to the same unit as the original data, which helps make the typical size of prediction errors clearer. RMSSE, in contrast, is especially helpful when working with time series forecasting. It normalizes the prediction error by comparing it against the variation of a naïve forecast (e.g., using the last observed value as the prediction). The result is a small metric that allows for easier comparison across models and datasets. An RMSSE value less than one indicates that the model performs better than the naïve benchmark, while a value greater than one suggests otherwise. Overall, MSE and RMSE are widely applied in general regression tasks, whereas RMSSE is especially effective for assessing model accuracy in time series forecasting scenarios**.**

3. Results and Discussions

The total of the data row is 1907 for the training set, and the testing set has six data points. We split the data at the beginning of the modeling phase, so the model will not contain the information from the training set. To understand further about the data, the descriptive numbers are present in Table 1.

Table 1. Descriptive of Training Set

Mean	Variance	SD
0.258	0.346	0.588

Based on Table 1, we see that the mean value is lower than the variance, this characteristic could be called overdispersion. Overdispersion happens cause of a lot of zero values in the data, the zero values represent nothing to be sold on the current day. Yet, when the sale happens, the value is quite high and far from zero, like five items, and vice versa. To build the hybrid model, we need to create the regression model first.

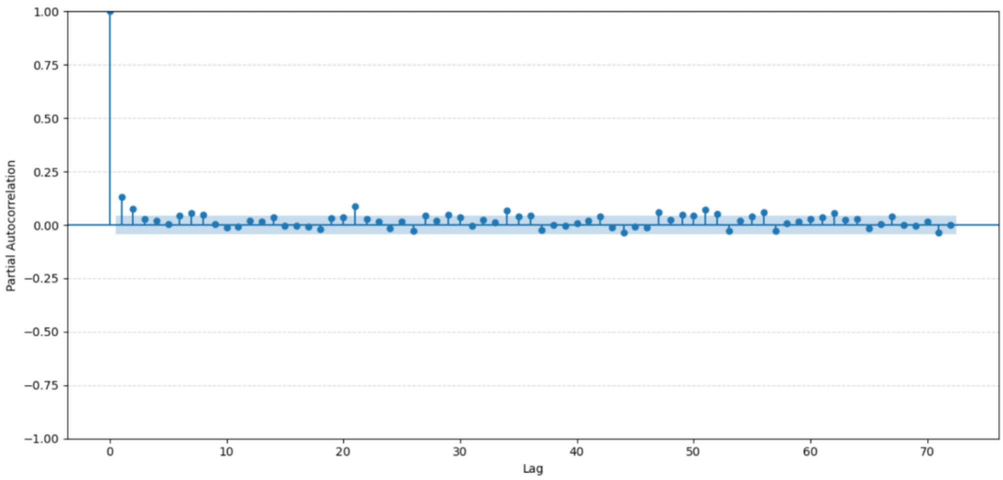

Figure 2. Partial Autocorrelation Function Analysis

DOI: https://doi.org/10.52435/jaiit.v7i2.704

In building the time series regression part, lagged variables were utilized as predictors, a practice common to regression modeling time or sequential data. To identify which lags had the most impact, an autocorrelation analysis was performed, the results of which are shown in Figure 2. The autocorrelation plot showed several statistically significant lags, which are 1, 2, 21, 34, 36, 42, 49, 51, 52, and 56. These lags were chosen due to their high correlation with the target variable. These ten lagged variables were chosen as regression model input variables, the model can be written as (5).

$$\begin{aligned}\hat{y} = 0.104 + 0.093y_{t-1} + 0.048y_{t-2} + 0.075y_{t-21} + 0.065y_{t-34} + 0.051y_{t-36} \\ + 0.051y_{t-42} + 0.048y_{t-49} + 0.07y_{t-51} + 0.049y_{t-52} + 0.073y_{t-56}\end{aligned} \tag{5}$$

The result from regression analysis will then be merged with the nonlinear model, in this case, a neural network, creating a stronger and more precise hybrid model altogether. Yet before the neural network is applied to the residual values, a normality test is performed to validate that the residual still contains some information to extract from the data. This is a great technique and robust approach if the process generating the data has both linear trends and complications.

After the time series regression model has been built, it will produce the fitted values or predictions. From these predictions, residuals can be calculated as the difference between the actual observed values and the formula $\varepsilon = y - \hat{y}$. Then the residuals are also analyzed and modeled through a neural network methodology. For residual modeling, we use a multi-layer feedforward neural network. The input layer consists of seven nodes for the seven lagged residual values that had been found to be beneficial in order to capture time dependencies. This is followed by a hidden layer with four neurons that is intended to capture higher-order, nonlinear interactions between lagged inputs. The last output layer is a single neuron that generates the predicted residual value.

All of the network layers are utilizing the ReLU (Rectified Linear Unit) activation function, which is popular due to its simplicity and effectiveness at dealing with non-linearities. Also, the use of ReLU is a consideration based on the data, because the data always produces non-negative values. To produce a non-negative result, ReLU is the best activation function because the function is max $(0, x)$. Another parameter to adjust is the epoch and batch size. We use 100 epochs and 200 batch size. The data consists of 1914 series. A 200-batch size with 100 epochs is more than enough to train the model. With Adam optimization starting from a 0.001 learning rate, it will make the model learn effectively and efficiently. It is also noteworthy that the model does not employ any dropout or regularization techniques, meaning that the network is quite small and might not be overfitting, given the data used. Once the residual predictions have been obtained from the neural network, the values are then inserted back into the initial time series regression model, that is, Figure 1, to fine-tune and enhance the final prediction. To assess the performance of this hybrid model, it is evaluated using standard forecasting benchmarks, including comparison with the exponential smoothing and Croston Method, as shown in Table 2.

Table 2. Model Evaluation Results

Method	Data	MSE	RMSE	RMSSE
Croston Original	Training	0.337	0.581	-
	Testing	1.141	1.068	1.378
Croston SBA	Training	0.337	0.580	-
	Testing	1.148	1.071	1.382
Exponential Smoothing	Training	0.370	0.608	-
	Testing	1.015	1	1.280
Proposed Model	Training	0.354	0.549	-
	Testing	0.723	0.850	0.986

Based on Table 2, the proposed method successfully outperforms the baseline method on each evaluation size metric on the testing set. This is because the exponential smoothing method cannot capture sudden extreme phenomena, and the Croston method is just an ad-hoc model that forecasts the data based on the if-else condition. At the same time, the proposed method can overcome this problem. Artificial neural networks overcome zero values in the data by predicting their residual values so that the mixed results can approach the actual value.

Furthermore, during the training phase, the hybrid model shows a slightly better fit, achieving a lower Mean Squared Error (MSE) of 0.354, compared to 0.370 from the exponential smoothing, yet not the Croston method. However, the contrast result happens in the Root Mean Squared Error (RMSE), where the hybrid model records 0.549, outperforming the 0.608 obtained by the exponential smoothing, 0.581 by the Croston Original, and 0.580 by the Croston SBA. These results suggest that the hybrid model is more effective in capturing patterns within the training data, with smaller average prediction errors [23] [24]. Lastly, in the RMSSE metric, the proposed model outperforms both the Croston and Exponential smoothing with a 0.986 value compared to 1.378, 1.382, and 1.280, respectively. By this result, the proposed model produces a promising report compared to the traditional forecasting method for intermittent data. To summarize the results of the proposed method and benchmark model, we can see **Figure *1***Figure 3 and Figure 4 as a subjective report.

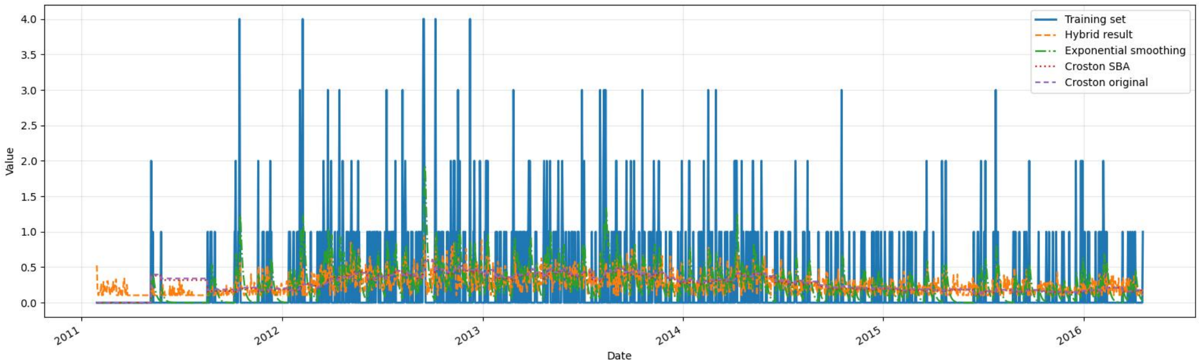

Figure 3. Forecast Performance Comparison Result on Training Set

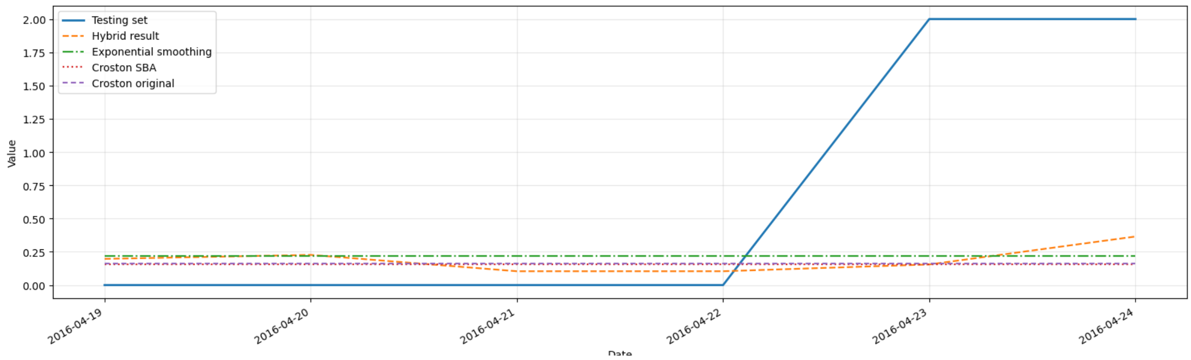

Figure 4. Forecast Performance Comparison Result on Testing Set

Figure 3 is a graphic that presents the training set compared to the fitted values produced by the Hybrid Model, Exponential Smoothing, and Croston Method. Actually, there is no significant difference between the three models, all of which produce the fitted number located among the mean value of the data. But the hybrid model can get the lowest difference between the ground truth, so that the metric evaluation comes the lowest among the models. With the same pattern as Figure 3, Figure 4 contains the same information. The hybrid model has the ability to adapt the zero pattern, the forecast result comes closest to the zero values of the ground truth. In this case, the performance of the forecasting model relies on averaging the forecasting results. Even though there is no exact forecast result that comes close to two, if we sum the forecasting value, it will get the same result as the total of the testing set.

4. Conclusion

The hybrid modeling technique that involves the use of artificial neural networks (ANN) with time series regression produces a promising outcome to forecast the intermittent data, especially those that

 DOI: https://doi.org/10.52435/jaiit.v7i2.704

are afflicted with a lot of zero values. Such data are known to have a distribution where the variance is higher than the mean, which led to the extreme event data. The traditional methods are less applicable to this current condition. By using a combination of linear and nonlinear modeling methodologies, the hybrid approach solves these problems better than a one-model framework.

In this hybrid framework, the time series regression component identifies the linear temporal patterns using lagged variables to build a strong foundation for identifying past trends. The ANN component, however, is used to identify the nonlinear pattern embedded in the residual patterns not captured by the regression model itself. Such complementarity enables the hybrid model to make more accurate predictions, as evidenced by the results of the evaluation. In comparison with the traditional exponential smoothing method and Croston method, the hybrid model performs better on all three performance metrics, which are MSE, RMSE, and RMSSE, especially in test data. All these findings indicate the strength and high capacity of the hybrid model in the ability to generalize to unseen data. Yet the forecasting value still does not represent the ground truth, but produces the lowest error.

In the future, the accuracy of forecasts can be further improved by adding distribution-based time series models, such as the Poisson or Negative Binomial distribution models. Through the adoption of a distribution that better approximates the underlying structure of the data, further research would have the potential to enhance model precision and adaptability, particularly with sparse or event-based time series data.

References

[1] S. Kolassa, "Commentary on the M5 forecasting competition," Int. J. Forecasting, vol. 38, no. 4, pp. 1562-1568, 2022, doi: 10.1016/j.ijforecast.2021.08.006

[2] X. Tian, H. Wang, and E. E, "Forecasting intermittent demand for inventory management by retailers: A new approach," J. Retail. Consum. Serv., 2021, doi: 10.1016/j.jretconser.2021.102662.

[3] A. Muhaimin, E. Setyowati, K. M. H, A. Ruhui, and F. Sari, "Intermittent Data Forecasting using Kernel Support Vector Regression," Int. Semin. Res. Mon. 2023. NST Proceedings., vol. 2024, pp. 23–26, 2024, doi: 10.11594/nstp.2024.4105.

[4] R. Sarlo, C. Fernandes, and D. Borenstein, "Lumpy and intermittent retail demand forecasts with score-driven models," Eur. J. Oper. Res., vol. 307, no. 3, pp. 1146-1160, 2023, doi: 10.1016/j.ejor.2022.10.006. Available: https://www.sciencedirect.com/science/article/pii/S0377221722007743.

[5] A. Muhaimin, P. Aji Riyantoko, H. Prabowo, and T. Trimono, "Negative Binomial Time Series Regression – Random Forest Ensemble in Intermittent Data," Int. J. Data Sci. Eng. Anaylitics, 2021, doi: 10.33005/ijdasea.v1i2.10.

[6] T. O. Hodson, "Root-mean-square error (RMSE) or mean absolute error (MAE): when to use them or not," Geosci. Model Dev., vol. 15, pp. 5481–5487, 2022, doi: 10.5194/gmd-15-5481-2022.

[7] S. Kim and H. Kim, "A new metric of absolute percentage error for intermittent demand forecasts," Int. J. Forecast., 2016, doi: 10.1016/j.ijforecast.2015.12.003.

[8] J. M. Rožanec, B. Fortuna, and D. Mladenić, "Reframing Demand Forecasting: A Two-Fold Approach for Lumpy and Intermittent Demand," Sustain., 2022, doi: 10.3390/su14159295.

[9] A. Muhaimin, D. D. Prastyo, and H. H. S. Lu, "Forecasting with recurrent neural network in intermittent demand data," in Proceedings of the Confluence 2021: 11th International Conference on Cloud Computing, Data Science and Engineering, 2021. doi: 10.1109/Confluence51648.2021.9376880.

[10] P. Liu, "Intermittent demand forecasting for medical consumables with short life cycle using a dynamic neural network during the COVID-19 epidemic," Health Informatics J., 2020, doi: 10.1177/1460458220954730.

[11] P. G. Zhang, “Time series forecasting using a hybrid ARIMA and neural network model,” Neurocomputing, 2003, doi: 10.1016/S0925-2312(01)00702-0.

[12] Suhartono, A. Choiruddin, H. Prabowo, and M. H. Lee, “Hybrid Machine Learning for Forecasting and Monitoring Air Pollution in Surabaya,” in Communications in Computer and Information Science, 2021. doi: 10.1007/978-981-16-7334-4_27.

[13] N. A. B. Kamisan, M. H. Lee, S. Suhartono, A. G. Hussin, and Y. Z. Zubairi, “Load forecasting using combination model of multiple linear regression with neural network for Malaysian City,” Sains Malaysiana, 2018, doi: 10.17576/jsm-2018-4702-25.

[14] S. Sharma, S. Sharma, and A. Athaiya, “Activation Functions In Neural Networks,” Int. J. Eng. Appl. Sci. Technol., 2020, doi: 10.33564/ijeast.2020.v04i12.054.

[15] Suhartono, P. D. Saputri, F. F. Amalia, D. D. Prastyo, and B. S. S. Ulama, “Model selection in feedforward neural networks for forecasting inflow and outflow in Indonesia,” in Communications in Computer and Information Science, 2017. doi: 10.1007/978-981-10-7242-0_8.

[16] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, “M5 accuracy competition: Results, findings, and conclusions,” Int. J. Forecast., 2022, doi: 10.1016/j.ijforecast.2021.11.013.

[17] R. Shah, P. Shah, C. Joshi, R. Jain and R. Nikam, "Linear Regression vs LSTM for Time Series Data," 2022 IEEE World Conference on Applied Intelligence and Computing (AIC), Sonbhadra, India, 2022, pp. 670-675, doi: 10.1109/AIC55036.2022.9848887.

[18] É. Perthame, F. Forbes, B. Olivier, and A. Deleforge, "Non linear robust regression in high dimension," in Proc. [Conference Name], 2016. Available: https://api.semanticscholar.org/CorpusID:36302546.

[19] S. Sharma, S. Sharma, and A. Athaiya, "Activation functions in neural networks," Int. J. Eng. Appl. Sci. Technol., vol. 4, pp. 310-316, May 2020, doi: 10.33564/IJEAST.2020.v04i12.054.

[20] Suhartono, M. H. Lee, and D. D. Prastyo, “Two levels ARIMAX and regression models for forecasting time series data with calendar variation effects,” in AIP Conference Proceedings, 2015. doi: 10.1063/1.4937108.

[21] R. J. Hyndman and A. B. Koehler, “Another look at measures of forecast accuracy,” Int. J. Forecast., 2006, doi: 10.1016/j.ijforecast.2006.03.001.

[22] W. Wei, Time Series Analysis: Univariate and Multivariate Methods Second Edition. United States of America: Pearson Education, Inc., 2006.

[23] H. Narayanan, T. Seidler, M. F. Luna, M. Sokolov, M. Morbidelli, and A. Butté, "Hybrid models for the simulation and prediction of chromatographic processes for protein capture," J. Chromatogr. A, vol. 1650, p. 462248, 2021, doi: 10.1016/j.chroma.2021.462248.

[24] N. G. Reich, J. Lessler, K. Sakrejda, S. A. Lauer, S. Iamsirithaworn, and D. A. T. Cummings, "Case study in evaluating time series prediction models using the relative mean absolute error," The Amer. Stat., vol. 70, no. 3, pp. 285–292, 2016, doi: 10.1080/00031305.2016.1148631.

 DOI: https://doi.org/10.52435/jaiit.v7i2.704