Journal of Applied Mathematics and Physics, 2024, 12, 543-553
https://www.scirp.org/journal/jamp
ISSN Online: 2327-4379
ISSN Print: 2327-4352

Prediction of Wordle Scores Based on ARIMA and LSTM Models

Biyun Chen, Wenqiang Li

College of Information Engineering, Yancheng Teachers University, Yancheng, China
Email: chenby@yctu.edu.cn

How to cite this paper: Chen, B.Y. and Li, W.Q. (2024) Prediction of Wordle Scores Based on ARIMA and LSTM Models. *Journal of Applied Mathematics and Physics*, **12**, 543-553.
https://doi.org/10.4236/jamp.2024.122036

Received: January 8, 2024
Accepted: February 26, 2024
Published: February 29, 2024

Abstract

This paper examines the effectiveness of the Differential autoregressive integrated moving average (ARIMA) model in comparison to the Long Short Term Memory (LSTM) neural network model for predicting Wordle user-reported scores. The ARIMA and LSTM models were trained using Wordle data from Twitter between 7th January 2022 and 31st December 2022. User-reported scores were predicted using evaluation metrics such as MSE, RMSE, R2, and MAE. Various regression models, including XG-Boost and Random Forest, were used to conduct comparison experiments. The MSE, RMSE, R2, and MAE values for the ARIMA(0,1,1) and LSTM models are 0.000, 0.010, 0.998, and 0.006, and 0.000, 0.024, 0.987, and 0.013, respectively. The results indicate that the ARIMA model is more suitable for predicting Wordle user scores than the LSTM model.

Keywords

Time Series, ARIMA, LSTM, Wordle, Prediction

1. Introduction

Wordle Puzzle is a popular game offered by the New York Times [1]. The objective of the game is to guess a 5-letter word in 6 or fewer attempts, with feedback provided for each guess. This study collected Wordle user scores from Twitter submissions between 7 January 2022 and 31 December 2022 through data mining [2]. Research on Wordle puzzle games typically focuses on problem-solving methods and game mechanism analysis [3] [4]. This paper analyses the scores reported by Wordle puzzle users using ARIMA and LSTM models for prediction. Firstly, the datasets need to undergo data cleaning to remove missing values and outliers. Additionally, the language should be clear, objective, and value-neutral, avoiding biased or emotional language. Finally, the text should be

free from grammatical errors, spelling mistakes, and punctuation errors. This paper aims to build ARIMA and LSTM models for predictive analysis of the cleaned data, as well as multiple regression models for comparative experiments. The models will be evaluated using quantitative metrics such as Root Mean Square Error (RMSE), Mean Square Error (MSE), R2, and Mean Absolute Error (MAE). It is important to maintain a clear and logical structure throughout the text, using simple sentences and avoiding complex terminology. The text should adhere to style guides and maintain consistent formatting, including citation and footnote styles.

2. Wordle Score Prediction Model

2.1. LSTM

LSTM neural network is an extension of Recurrent Neural Network (RNN), which solves the problem of long-term dependency [5] [6]. The basic structure of an LSTM unit consists of forgetting gates, input gates and output gates, and the gates implement the function of forgetting or remembering, and the basic structure of the unit is shown in Figure 1.

The forgetting gate takes the inputs of the current moment and the outputs of the previous moment as inputs to a sigmoid function. This function controls the extent to which the state of the previous unit has been forgotten. The input gate is composed in combination with the tanh function and is used to control the amount of new input information. The output layer determines the output information by processing the current cell state using the tanh function. The weights obtained from the sigmoid function are then combined to filter some of the cell state information and obtain the output for the next moment.

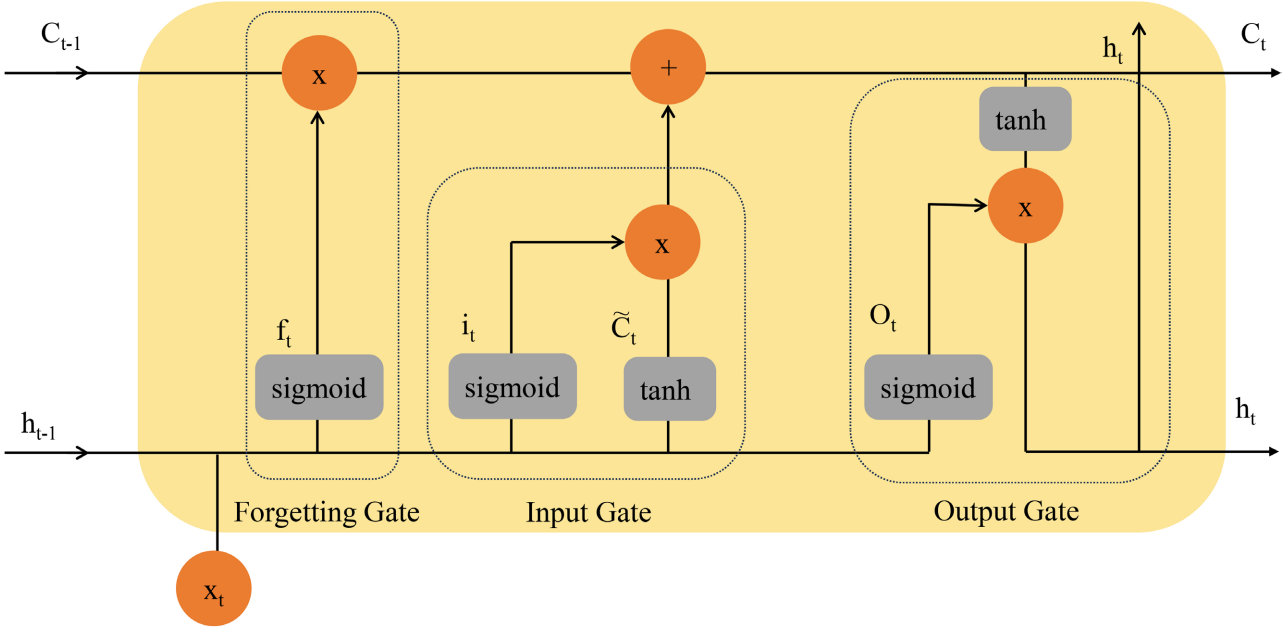

Figure 1. The basic structure of an LSTM cell.

$$f_t = \sigma\left(W_f \times \left[h_{(t-1)}, x_t\right] + b_f\right) \tag{1}$$

$$i_t = \sigma\left(W_i\left[h_{(t-1)}, X_t\right] + b_i\right) \tag{2}$$

$$\tilde{C}_t = \tanh\left(W_c\left[h_{(t-1)}, X_t\right] + b_c\right) \tag{3}$$

$$O_t = \sigma\left(W_o\left[h_{(t-1)}, X_t\right] + b_o\right) \tag{4}$$

$$C_t = f_t * C_{(t-1)} + i_t * \tilde{C}_t \tag{5}$$

$$h_t = O_t * \tanh C_t \tag{6}$$

In Equations (1)-(6), f_t is the output of the Oblivion gate; i_t is the output of the input gate; O_t is the output of the output gate; $\tilde{C}_t$ is the current input memory; $C_{(t-1)}$ is the cell loading at the previous moment; C_t is the cell state at the current moment; $h_{(t-1)}$ is the output of the current moment; W_f, W_i, W_o, W_c are the weights of the forgetting gate, the input gate, the output gate, and the input gate intermediate variable X_t multiplied with the input at the current moment and the output at the previous moment $h_{(t-1)}$, respectively; b_f, b_i, b_o, b_c are bias vectors; σ is the sigmoid function. The process of LSTM modelling is divided into three steps. Firstly, the sample data that will enter the input layer undergoes data normalisation. Then, the data that meets the LSTM input requirements is input into the hidden layer. The implicit layer produces multiple results which are then mapped in the output layer to generate the desired model results. The model is then trained using safety accident data over a set iteration period to predict the trend of safety accidents. Finally, the trained model is used to predict and analyze the test set data, and the model's fitting effect is assessed by calculating the error function.

2.2. The ARIMA Model

Time series mainly include smooth time series models of autoregressive (AR) [7], moving average (MA) and autoregressive moving average (ARMA) [8] models, and non-smooth time series models of differential ARIMA [9] [10]. Time series models play a key role in performing time series analysis to represent the characteristics of a time series. The values at each time represent the observations of a phenomenon at that time, where neighbouring time intervals can be different. Assuming a time series, Equation (7).

$$X = \left\{\left(t_1, x_1\right), \cdots, \left(t_i, x_i\right), \cdots, \left(t_n, x_n\right)\right\},\ t_i < t_{i+1}\left(i = 1, \cdots, n-1\right) \tag{7}$$

where, t_i represents time, x_i represents the observed value, $\left(t_i, x_i\right)$ represents the observed value x_i at time t_i.

ARIMA(p,d,q), AR is "autoregressive", p is the number of autoregressive terms; MA is "sliding average", q is the number of sliding average terms, and d is the number of differences (order) made to make it a smooth series. The number of differences (orders). The non-smooth time series for the d-order difference processing, the first to make it into a smooth series, and then its data into the

ARMA model for fitting, abbreviated as ARIMA(p,d,q), see formula (8):

$$x_t = \varphi_1 x_{t-1} + \cdots + \varphi_p x_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} \tag{8}$$

where $\varphi_1, \varphi_2, \cdots, \varphi_p$ the autoregressive order, $\theta_1, \theta_2, \cdots, \theta_q$ is the moving average coefficient, q is the moving average order, p is the autoregressive order, and ε_t is the white noise process.

3. Modelling and Solving

3.1. Data Preprocessing

In this paper, the scores of user-submitted reports from Twitter from 7 January 2022 to 31 December 2022 are selected as a datasets, with a time interval of nearly one year, which is sizable in terms of the time distribution, and able to present a relatively easy-to-observe distribution of user-reported scores over the time series. The data contains both time series and user report scores, and a preliminary state distribution plot of the growth of report scores over the time series is shown in Figure 2.

Firstly, the datasets undergoes data cleaning to ensure accuracy. Abnormal sample data is manually filled in a reasonable manner to balance the data samples. Next, the overall trend, seasonal trend, and residual distribution of the data are observed as shown in Figure 3. In this paper, we analysed the series using the seasonal_decompose function in the statsmodels library. The seasonal distribution shows that the seasonal indices of the series are all approximately 1, indicating no significant seasonality. The residual values are also approximated to. The scores reported by the users indicate an upward trend from January 2022 until the start of February 2022, reaching their highest point at the beginning of February 2022, followed by a downward trend.

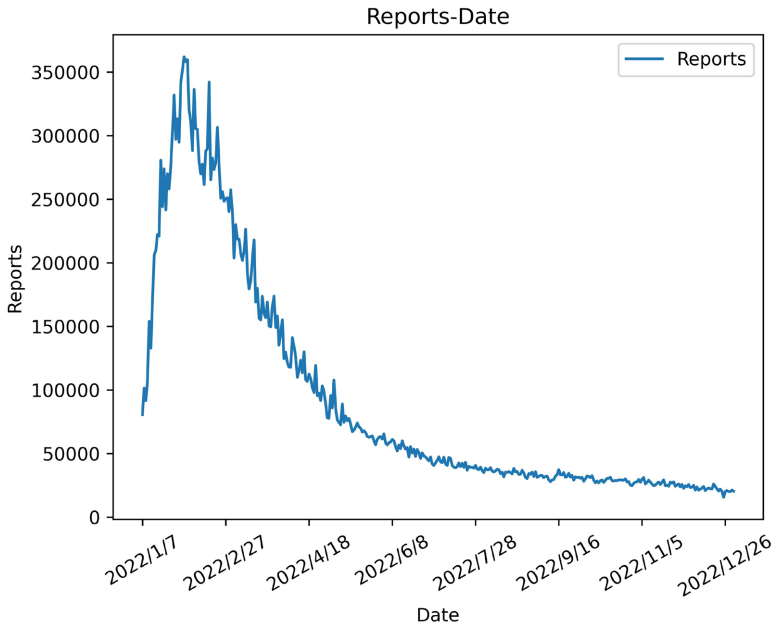

Figure 2. Report-line chart of date distribution.

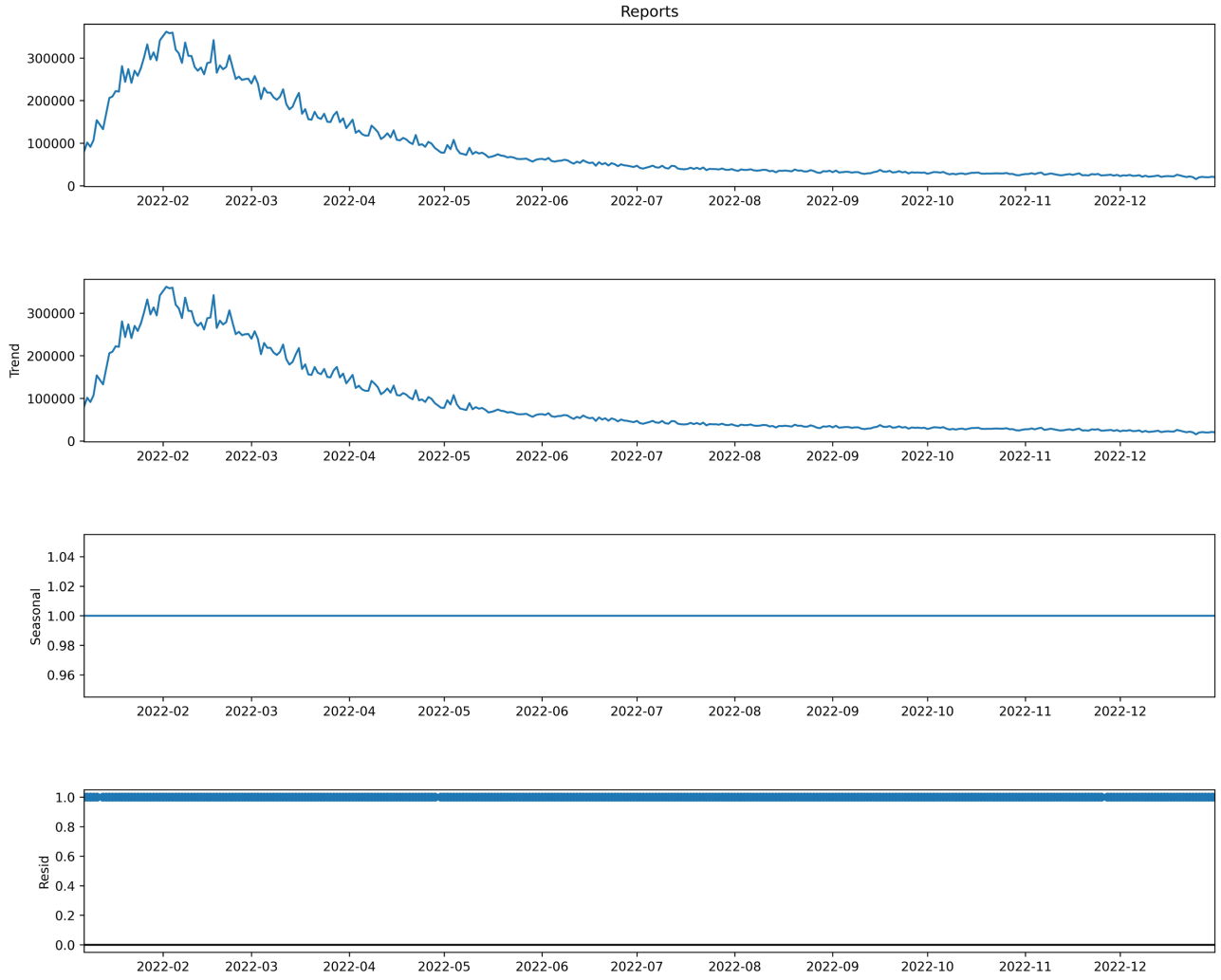

Figure 3. Seasonal decomposition diagram.

Before model training, to better fit the data, normalize data. Its function is to scale the current value using the maximum and minimum values of the data, so that the value of the data is between [0, 1], and the normalization formula is see formula (9):

$$x_i' = \frac{x_i - \min(x)}{\max(x) - \min(x)} \tag{9}$$

In formula. (9), x_i represent the unprocessed data, $\min(x)$ represent the overall minimum data and $\max(x)$ represent the overall maximum data.

3.2. ARIMA Time-Series Prediction

To ensure the model can process the time series data, it is crucial to first determine its stability. This paper tests the data smoothness using various methods. The datasets in this paper appear to be relatively smooth, as shown in **Figure 4**. The rolling().mean() and rolling().std() functions were then used to calculate the rolling mean and standard deviation, respectively, without using the time period.

The resulting fitting graphs are displayed in Figure 4.

From Figure 4 it can be seen that the rolling average fits very well and the sequence is stable.

Next, this paper makes a judgement through the unit root test (ADF test). This method determines whether the series is smooth or not by looking at the presence of a unit root, *i.e.*, the hypothesis of the test is the presence of a unit root, and by looking at the significance test statistic to see if it is less than the three confidence levels (10%, 5%, 1%). By performing ADF test on the standardised data, the p-value of the ADF test is 0.001485 through Table 1 and the significant level is generally 0.05, therefore the p-value is less than the significant level and the original hypothesis is rejected. The data is stable as the values of Test Static Value are less than the values at all three confidence levels.

After ADF test, the data is a smooth series. The white noise test is performed on the sequence and two values of statistic and p-value are obtained which are 351.092166 and 2.450789e−78 respectively, it can be concluded that the p-value is significantly less than the significant level and hence the time series is a smooth non-white noise series.

The model parameters are determined by analysing the p and q parameters in the selected range shown in Figure 5. Each of the selected parameters is then evaluated, and based on the results, the one-parameter model with the best performance is chosen. The evaluation criteria are based on the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC). Table 2 presents the results of intercepting some of the parameters with better outcomes and determining the use of parameter (1,1,1) for model fitting. The model fitting graph is displayed in Figure 6.

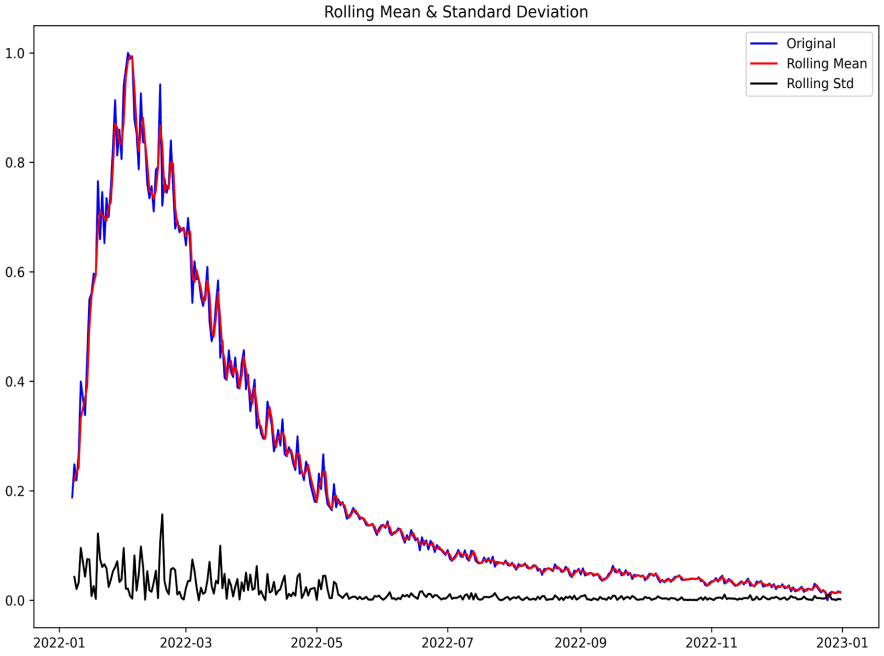

Figure 4. Rolling mean & standard deviation.

Table 1. ADF test results for time-series data.

Adfuller test p-value	Reports
Test Statistic Value	−3.986037
p-value	0.001485
Lags Used	17
Number of Observations Used	338
Critical Value (1%)	−3.449846
Critical Value (5%)	−2.870129
Critical Value (10%)	−2.571346

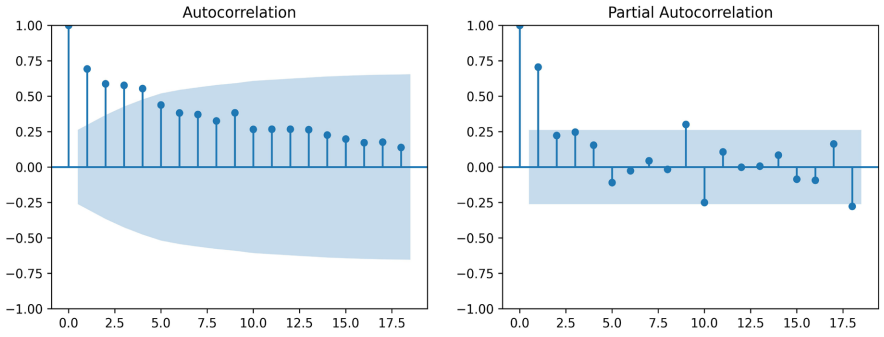

Figure 5. Autocorrelation diagram and partial autocorrelation diagram.

Table 2. Some parameter results.

Model parameter	AIC	BIC
(1,1,0)	−1382.878445242443	−1375.4849294035548
(0,1,1)	−1383.2291649825056	−1371.2620918740167
(1,1,1)	−1382.878445242443	−1371.2620918740167
(0,1,2)	−1383.2291649825056	−1366.2340172110678

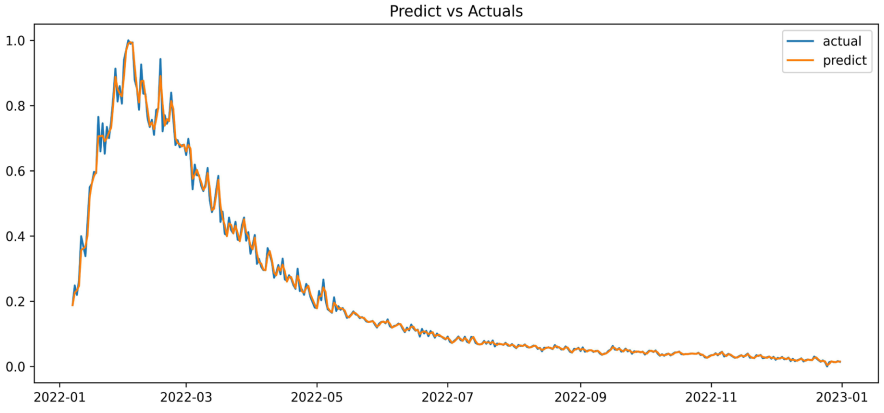

Figure 6. ARMA(1,1,1) fitting diagram.

3.3. LSTM Prediction

The LSTM model was constructed with a sequence length of 30 and a sequence element dimension of 1. The hidden layer contains 50 neurons and there is only one fully linked layer. The neuron output weight is set to 0.2 and the intermediate state of returning to the LSTM is set to True. The optimal hyperparameters were selected through grid search. The batch size of the input data is set to 16 and the model was trained on 8 data points for 30 days using the loss function "mse" for 50 iterations. The batch size for input data was set to 16, with 8 data points selected for training. The training period lasted for 30 days, with 50 training iterations. The loss function used was "MSE". The model was fine-tuned using the Adam optimizer and built using TensorFlow.

3.4. Evaluation Indicators

To evaluate the predictive performance of the model, this paper uses MSE, RMSE, R^2, and MAE as evaluation metrics. MSE measures the expected value of the squared difference between the predicted and actual values. The accuracy of the model increases as the value of RMSE decreases, as it is the square root of MSE. R^2 compares the predicted values with the mean only, and the closer the result is to 1, the more accurate the model is. MAE is the average of the absolute errors between the predicted and actual values, and the closer the value is to 0, the more accurate the prediction.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(\hat{y}_i - y_i\right)^2 \tag{10}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\hat{y}_i - y_i\right)^2} \tag{11}$$

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|\hat{y}_i - y_i\right| \tag{12}$$

$$\mathrm{R}^2 = 1 - \frac{\sum_i \left(\hat{y}^{(i)} - y^{(i)}\right)^2}{\sum_i \left(\bar{y} - y^{(i)}\right)^2} = 1 - \frac{\left(\sum_{i=1}^{m}\left(\hat{y}^{(i)} - y^{(i)}\right)^2\right)\Big/ m}{\left(\sum_{i=1}^{m}\left(y^{(i)} - \bar{y}\right)^2\right)\Big/ m} = 1 - \frac{\mathrm{MSE}\left(\hat{y}, y\right)}{\mathrm{Var}\left(y\right)} \tag{13}$$

where, $\hat{y}_i$ represent the predicted value, y_i represent the true value, m represent the number of times.

3.5. Results Analysis

To enhance the analysis and comparison of ARIMA and LSTM models in predicting Wordle user-reported scores, this paper constructs several regression models using the same data for prediction. Table 3 presents the metrics of the model's results for predicting Wordle user-reported scores in 2022.

Table 3 displays the prediction results of ARIMA, LSTM, and multiple regression models. The ARIMA(0,1,1) model has the best prediction accuracy with

an MSE, RMSE, R^2, and MAE of 0.000, 0.010, 0.998, and 0.006, respectively, followed by LSTM. To enable a more intuitive comparison, we will compare the ARIMA(0,1,1) model and the LSTM model separately.

Figure 7 and Figure 8 shows a comparison of the two models and the actual values. From the Figure 7 and Figure 8, it is evident that both models can fit the trend of the number of user reports on the curve. It is evident that the LSTM model's fitting accuracy for the effect of the ARIMA(0,1,1) is inferior compared to it, and there is still a significant difference between the predicted and actual values.

Table 3. Prediction results of the model.

Model	MSE	RMSE	R^2	MAE
ARIMA(1,1,0)	0.000	0.013	0.997	0.007
ARIMA(0,1,1)	**0.000**	**0.010**	**0.998**	**0.006**
ARIMA(1,1,1)	0.000	0.013	0.997	0.007
LSTM	0.000	0.024	0.987	0.013
LightBGM	0.001	0.029	0.989	0.016
XGBoost	0.002	0.047	0.970	0.029
GBDT	0.002	0.041	0.982	0.027
AdaBoost	0.002	0.046	0.975	0.029
Random Forest	0.001	0.024	0.990	0.014
Decision Tree	0.001	0.023	0.984	0.014

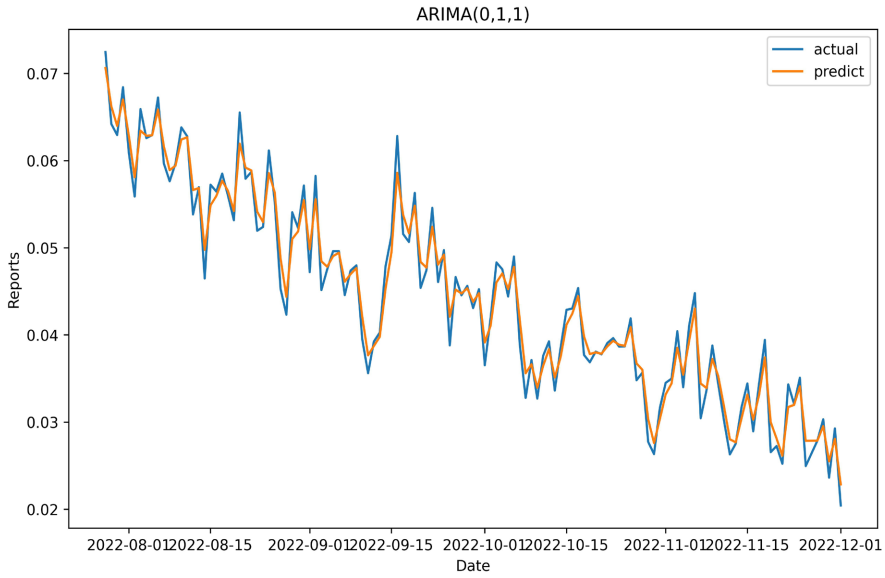

Figure 7. The prediction reports by ARIMA(0,1,1) model.

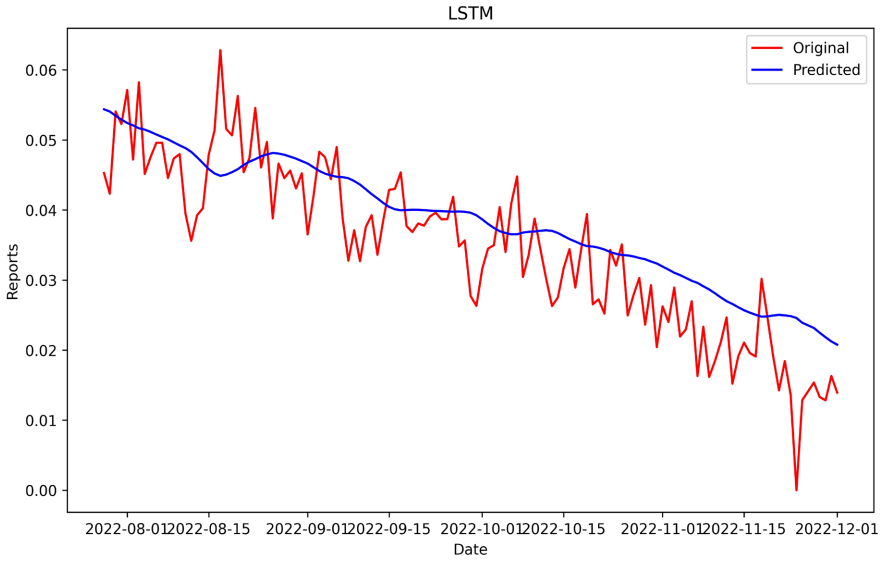

Figure 8. The prediction reports by the LSTM model.

4. Conclusion

This paper presents a comparative study of ARIMA and LSTM models for predicting Wordle user scores. Both models performed well in fitting and predicting user-reported scores. However, the ARIMA model was found to be more accurate than the LSTM model. It is worth noting that the ARIMA model required more steps for data processing than the LSTM model. The model presented in this paper has some shortcomings. The ARIMA model of ACF and PACF fixed order did not achieve the best score, indicating unresolved issues in the model's establishment process. To address this, the paper will continue to improve the existing model and determine the values of p and q by performing the difference.

Data Availability Statement

The data that support the findings of this study are available from the first author upon reasonable request.

Funding

This research was funded by 2023 General Project of Educational Teaching Reform Subjects in Yancheng Teachers University (Grant No. 2023YCTCJGY34), Yancheng Teachers University 2023 Curriculum Civics Demonstration Course Project.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Shi, L., Chen, Y., Lin, J., *et al.* (2023) A Black-Box Model for Predicting Difficulty of

Word Puzzle Games: A Case Study of Wordle. *Knowledge and Information Systems*, **66**, 1729–1750. https://doi.org/10.1007/s10115-023-01992-6

[2] Wordle-The New York Times (2022). https://www.nytimes.com/games/wordle/index.html

[3] Hefkaluk, N., Linehan, C. and Trace, A. (2024) Fail, Fail Again, Fail Better: How Players Who Enjoy Challenging Games Persist after Failure in "Celeste". *International Journal of Human-Computer Studies*, **183**, Article 103199. https://doi.org/10.1016/j.ijhcs.2023.103199

[4] Sun, C.T., Wang, D.Y. and Chan, H.L. (2011) How Digital Scaffolds in Games Direct Problem-Solving Behaviors. *Computers & Education*, **57**, 2118-2125. https://doi.org/10.1016/j.compedu.2011.05.022

[5] Yu, Y., Si, X., Hu, C., *et al*. (2019) A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. *Neural Computation*, **31**, 1235-1270. https://doi.org/10.1162/neco_a_01199

[6] Staudemeyer, R.C. and Morris, E.R. (2019) Understanding LSTM—A Tutorial into Long Short-Term Memory Recurrent Neural Networks.

[7] Cheng, J.S., Yu, D.J. and Yu, Y. (2006) A Fault Diagnosis Approach for Roller Bearings Based on EMD Method and AR Model. *Mechanical Systems and Signal Processing*, **20**, 350-362. https://doi.org/10.1016/j.ymssp.2004.11.002

[8] Makridakis, S. and Hibon, M. (1997) ARMA Models and the Box-Jenkins Methodology. *Journal of Forecasting*, **16**, 147-163. https://doi.org/10.1002/(SICI)1099-131X(199705)16:3<147::AID-FOR652>3.0.CO;2-X

[9] Duan, G., Su, Y. and Fu, J. (2023) Landslide Displacement Prediction Based on Multivariate LSTM Model. *International Journal of Environmental Research and Public Health*, **20**, Article 1167. https://doi.org/10.3390/ijerph20021167

[10] Hyndman, R.J. and Khandakar, Y. (2008) Automatic Time Series Forecasting: The forecast Package for R. *Journal of Statistical Software*, **27**, 1-22. https://doi.org/10.1080/10919390802199012