Analysis of Wordle Game Mechanism Based on LSTM and MLP

Yixi Zhou [1,*], Hanyang Cao [2], Xuanbo Jia [3]

[1] Department of Intelligent science and technology, Hangzhou Dianzi University, Hangzhou, China

[2] School of Economics, Hangzhou Dianzi University, Hangzhou, China

[3] School of Computing, Hangzhou Dianzi University, Hangzhou, China

* Corresponding Author Email: author@abcd.edu

Abstract. This paper uses data analysis, deep learning and natural language processing techniques to study the difficulty and result distribution of word guessing based on the Wordle game mechanism. First, a three-layer LSTM model is established to predict the number of reported results, and the correlation between word factors and guessing difficulty is analyzed. Next, a multi-sequence LSTM model combined with NLP models is established to predict the result distribution of specific words at specific times. Finally, an MLP classification model is built to classify the difficulty of words.

Keywords: Three-layer LSTM, multiple sequence prediction model, NLP model, MLP model.

1. Introduction

Wordle is a popular puzzle currently offered daily in the New York Times. The game allows players to solve the puzzle by guessing a five-letter word six times or less, with each guess receiving feedback and judgment from the system: green for a correct letter and correct position, yellow for a correct letter but wrong position, and gray for a wrong letter. For the current version, each guess must be a real English word, i.e., the word exists in the lexicon as determined by the contest, otherwise the input is judged invalid.

In addition, players can choose to play in either the normal mode or the hard mode, with the hard model limiting players to the next round of guessing using the known correct letters on top of the normal mode. In order to obtain the distribution of users' game play, MCM crawled through users' shared game information on Twitter to obtain data from January 7, 2022 to December 31, 2022, from which all the content analysis in this paper was conducted.

2. LSTM prediction and results for game report data results

2.1. Prediction principles and steps

Given the necessity of handling lengthy time series data to capture more temporal dependencies, coupled with the objective of training a predictive model that demonstrates strong accuracy and generalization capabilities, we propose the adoption of a three-layer LSTM network to process the sequence data and achieve the prediction of report data results. The three-layer LSTM shares a similar basic structure to the standard LSTM, encompassing an input layer that accepts normalized time series data N_{norm}, three LSTM layers, and a fully connected output layer. We define the following variables:

$$x_t = N_{normt} \tag{1}$$

$$h_t^{(1)} = \mathrm{LSTM}\left(x_t, h_{t-1}^{(1)}\right) \tag{2}$$

$$h_t^{(2)} = \mathrm{LSTM}\left(h_t^{(2)}, h_{t-1}^{(2)}\right) \tag{3}$$

$$h_t^{(3)} = \mathrm{LSTM}\left(h_t^{(3)}, h_{t-1}^{(3)}\right) \tag{4}$$

Where $h_t^{(1)}$ is the hidden state of the first layer LSTM, $h_t^{(2)}$ is the hidden state of the second layer LSTM, $h_t^{(3)}$ is the hidden state of the third layer LSTM, and x_t is the input of time step t, i.e., the normalized time series data N_{norm}, $h_{t-1}^{(1)}$, $h_{t-1}^{(2)}$ and $h_{t-1}^{(3)}$ are the hidden states of the first, second and third layer LSTMs at time step $t-1$, respectively.

After the third layer LSTM output, $h_t^{(3)}$ is output via the fully connected layer, the final variable y_t

$$y_t = W_{out} h_t^{(3)} + b_{out} \tag{5}$$

Where W_{out} and b_{out} denote the weight and bias parameters of the output layer, respectively. The output sequence $N_{predict,norm}$ can be obtained from $N_{predict,norm,t} = y_t$, where y_t represents the predicted output variable for the input variable at time step t. The reverse normalization of the output sequence can be performed using the standard deviation method to obtain the final prediction results.Figure 1 is the model architecture diagram of the three-layer LSTM.Therefore, in this paper, the LSTM model is used to predict the report result data on March 1, 2023.

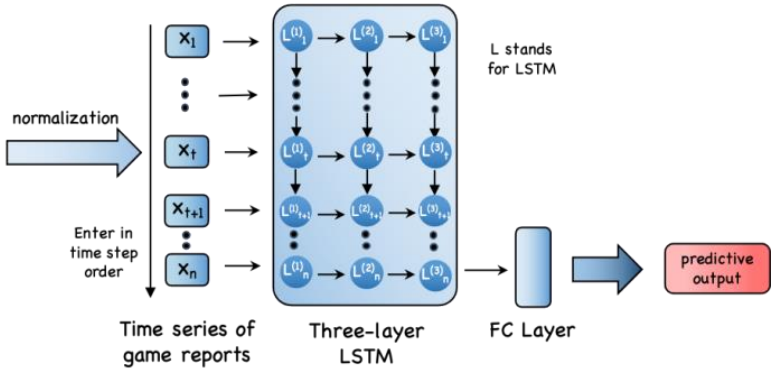

Figure 1. the reports number prediction model based on a three-layer LSTM

2.2. Prediction results and tests

This paper constructed and trained an LSTM model to predict the test set, and calculated the mean squared error (MSE) between the predicted and actual results. Finally, the original data, predictive training data and predictive test data are visualized on Figure 2, and the predicted data for March 1, 2023 and the test set MSE were output. The predicted value for March 1, 2023 is 24266.6847.

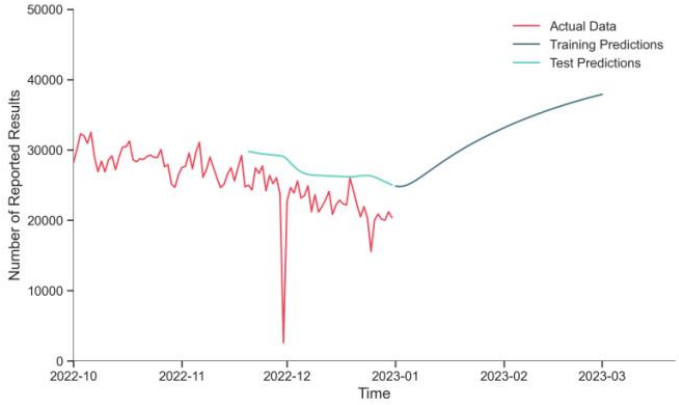

Figure 2. Details of prediction results

The Mean Squared Error (MSE) is a measure of the difference between the predictions of a regression model and the actual observed values. It is calculated as the average of the squared differences between the predicted values and the actual values, according to the following formula:

$$MSE = \frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y_i})^2 \tag{6}$$

Where n is the number of observations, y_i represents the true value of the ith observation, and $\widehat{y_i}$ represents the predicted value of the ith observation. A smaller MSE indicates that the model's predictions are closer to the actual observed values.

MSE of the test set: 0.000236

MSE of the training set: 0.000583

As can be seen from Figure 3, the LSTM model trained in this paper has better prediction effect.

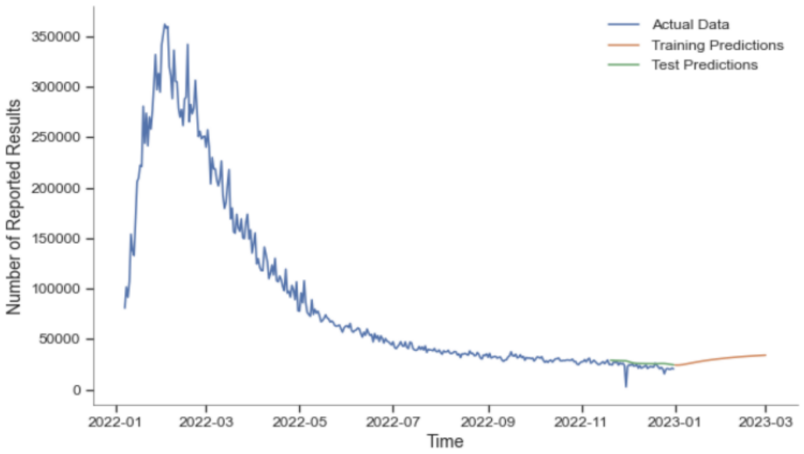

Figure 3. Overall predicted results

2.3. Word feature attribute extraction

According to some common characteristic laws of English linguistics and word spelling rules, the word sequences of the data set are analyzed and summarized, and the following types of word feature attributes can be obtained.They are Letter frequency characteristics (number of vowel letters possessed in a word),Common degree characteristics (calculate how often a word appears in the Brown corpus),Does it contain consecutive consonant letters,Does it contain consecutive vowel letters,Does it start with a vowel letter,Whether or not it ends in a vowel letter.For the relationship between the word feature attribute and the hardmode ratio, define the variable hardmode score ratio

$$D = \frac{N_h}{N_{sum}} \tag{7}$$

where N_h represents the difficulty scores of the word sequence in the data set, and N_{sum} represents the total scores obtained by the word sequence in the data set.

Pearson correlation test was performed on 6 word feature variables and variable difficulty mode scores, and the covariance and standard deviation between the two variables were calculated to measure their linear correlation. By normalization method, the data of the six types of characteristic variables and the difficult mode proportion variable D data for correlation coefficient testing are scaled to a range between 0 and 1. Finally, Table 1 of Pearson's correlation coefficient for the word feature variable and the hardmode score ratio variable is obtained.

Table 1. Pearson's correlation coefficients between the word feature variables and the difficulty mode score ratio variable D'

word	commonness	has consecutive vowels	starts with vowel	ends with vowel	total freq
number	-0.0112	-0.0069	0.0307	-0.0170	-0.0273

The correlation coefficients between the feature variables of each word and the variable D' of the difficulty pattern scores, as shown in the table, are all less than 0.03 in absolute value. Therefore we can conclude that there is no linear correlation between the word attribute and the percentage of registrations in the difficult mode based on this. We plot a scatter plot of the two variables and based on Figure 2, and we can infer that the word attribute does not affect the percentage of registrations in the difficult mode.

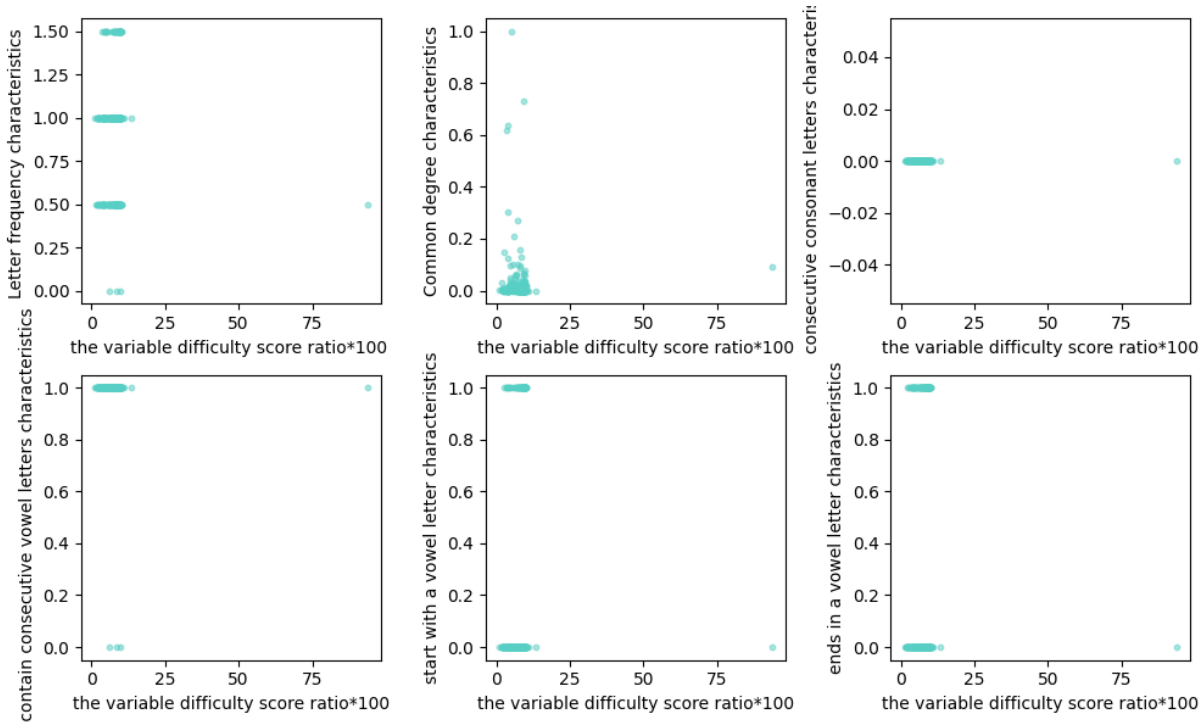

Figure 4. scatter plots of the relationship between feature values of each word and the variable difficulty score ratio *100

3. Multi-sequence LSTM model

3.1. Model building

First, the date data is converted to [year, month, day] temporal features and combined with word features to form a feature matrix. There are 13 features in the final feature matrix, the first 3 are temporal features and the next 10 are word features.

Table 2. Feature

Type	Property
date features	year
date features	month
date features	day
Letter frequency characteristics	$freq_{\ell}$
Common degree characteristics	commonness(w)
Does it consecutive consonant letters	hcc(w)
Does it contain consecutive vowel letters	hcv(w)
Does it start with a vowel letter	sv(w)
Whether or not it ends in a vowel letter	ev(w)

Therefore, when constructing the LSTM model,the input feature dimension is set to 8 and the MinMaxS- caler method is used to scale the data features to a specific range so that all feature values are between 0 and 1. The normalization can avoid the computational errors caused by too large or too small feature values, and can make the weights of different features more balanced.

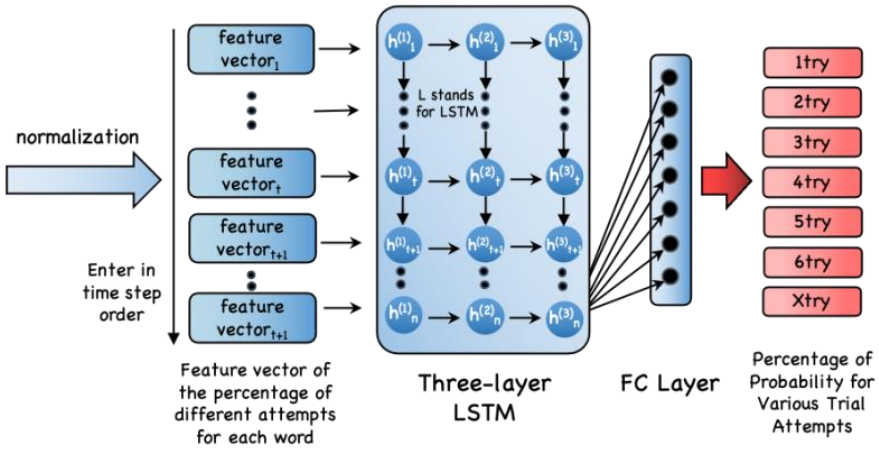

Figure 5. The pass rate prediction model based on a Multi-Sequence LSTM model

Define a three-tier LSTM model as shown in Figure 5, where the LSTM layers of the first and second layers return a sequence, and the output of the LSTM layer of the third layer is a vector with 7 elements, representing the probability of 7 guessed result distributions. This article normalizes the output using the softmax activation function. The loss function of the model is cross-entropy loss, the optimizer uses the Adam optimizer, and the evaluation metric is accuracy.

3.2. EERIE's predictions

Based on the specific word (EERIE) distribution predicted by the model at a specific time (March 1, 2023), the feature vector is converted into a shape suitable for input to the model. Add the feature vector of the specific word EERIE to the word feature list and set the date feature to the date value of March 1, 2023, according to our prediction result, the most likely number of guesses is three times, and the predicted result follows the following distribution.

Table 3. Distribution of the number of guesses for the word EERIE in 20230301

Try_number	1_try	2_try	3_try	4_try	5_try	6_try	X_try
Proportion	0.06	0.40	10.21	65.74	22.28	1.25	0.06

4. Build an MLP model and determine the difficulty of words

We select word frequency and temporal features to form feature vectors and classify words using multilayer perceptrons (MLPs). First, we calculate the difficulty score for each word based on the average number of guesses for each word in the historical time series and sort them in ascending order. We then divide words into three categories based on their scores: difficult, medium, and easy. Words with higher scores belong to the difficult category, while words with lower scores belong to the easy category.

4.1. Definition of word difficulty score

Define a difficulty score score based on the weighted average number of guesses $\overline{N_{try}}$ per word in the historical time series with the distribution of the number of failures

$$\overline{N_{try}} = \frac{\sum_{i=1}^{n} w_i N_{try,i}}{\sum_{i=1}^{n} w_i} \tag{8}$$

$$score = \frac{\overline{N_{try}}}{1-\frac{X_{try}}{100}} \tag{9}$$

The difficulty score of each word was calculated and ranked, and the words were then classified into difficult, moderate, and easy difficulty three categories. Words with higher difficulty scores belonged to the top 10% category of difficulty and words with lower difficulty scores belonged to

the bottom 10% category of easy. The lower the difficulty score, the easier the words are. The results of word classification were obtained, See Figure 6.

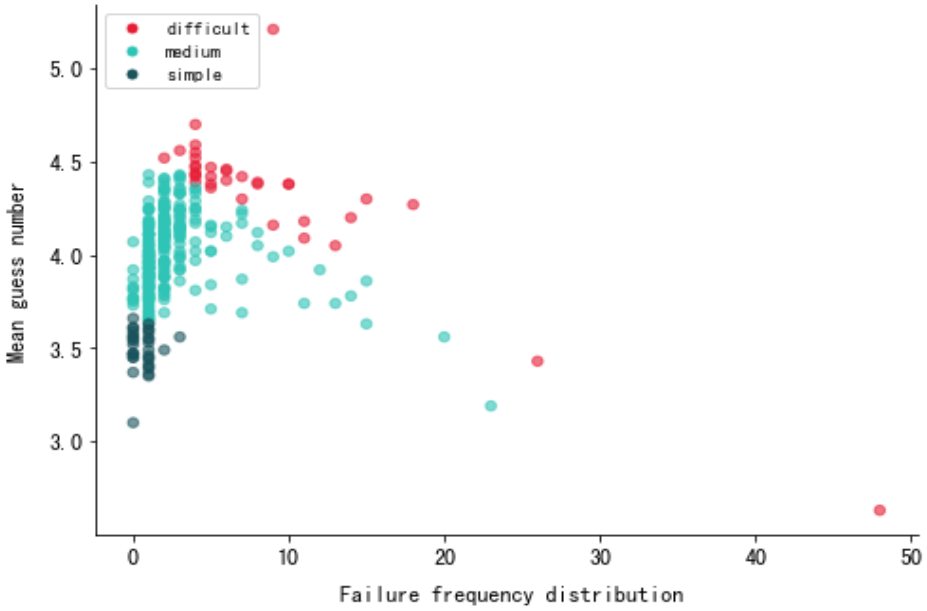

Figure 6. Scatter plot of category word distribution after classification

4.2. Classification model

A Multilayer Perceptron (MLP) is a neural network-based classification model that consists of multiple layers of neurons, each of which is connected to the next layer of neurons. During the training process, the multilayer perceptron adjusts the weights by a backpropagation algorithm to improve the accuracy of the classifier. In this problem, we construct a multilayer perceptron with two hidden layers, the first hidden layer contains 100 neurons and the second hidden layer contains 50 neurons. During the training process, we use a backpropagation algorithm to update the weight matrix and bias vector to enable the neural network to classify the input data more accurately. The following is the classification formula of the multilayer perceptron with two hidden layers built in this paper.

$$f(x) = o^{(L)} = \sigma\left(W^{(L)}\sigma\left(W^{(2)}\sigma\left(W^{(1)}x + b^{(1)}\right) + b^{(2)}\right) + b^{(L)}\right) \tag{10}$$

where the input sample x is the feature vector, $f(x)$ denotes the classification result of the input sample x, L denotes the output layer of the neural network, and σ denotes the activation function. $\mathbf{W}^{(l)}$ and $\mathbf{b}^{(l)}$ denote the weights and bias parameters of the lth layer, respectively. The $\mathbf{o}^{(L)}$ denotes the output of the neural network in the output layer with the size of the classification number C, i.e., $\mathbf{o}^{(L)} \in \mathbb{R}^{C}$. Finally, the output $f(x)$ of the output layer can be used as the prediction result for the calculation of the cross-entropy loss function L and the update of the model parameters according to the classification labels after classifying the word dataset above.

$$L = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{i,j}\log\left(\hat{y}_{i,j}\right) \tag{11}$$

where N denotes the number of samples, $y_{i,j}$ denotes the true value of the jth categorical label for the ith sample, and $\hat{y}_{i,j}$ denotes the predicted value of the jth categorical label for the ith sample.

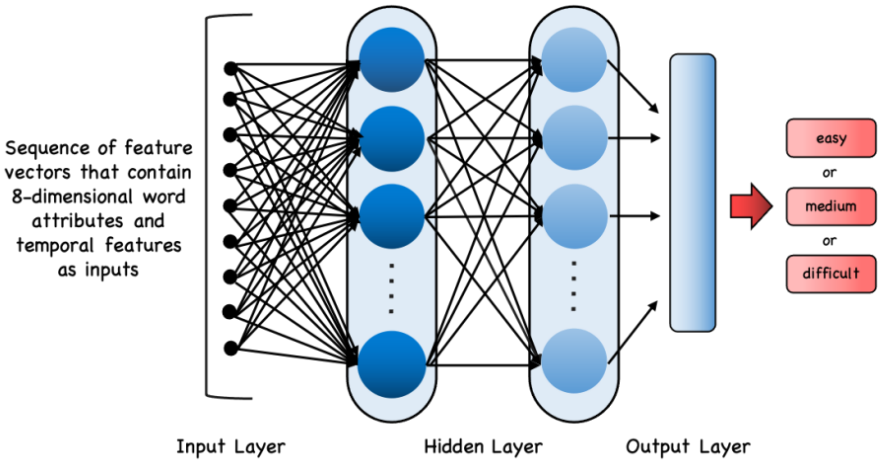

Figure 7. Details of prediction results

4.3. Multilayer Perceptron Model Evaluation

Table 4. Classification Report

	Precision	Recall	F1-score	Support
difficult	0.50	0.25	0.33	12
medium	0.74	0.86	0.80	73
simple	0.41	0.30	0.35	23
Accuracy: 67.59%				
Precision: 64.42%				
Recall: 67.59%				
F1-score: 0.65				
Macro avg: 0.55, 0.47, 0.49, 108				
Weighted avg: 0.64, 0.68, 0.65, 108				

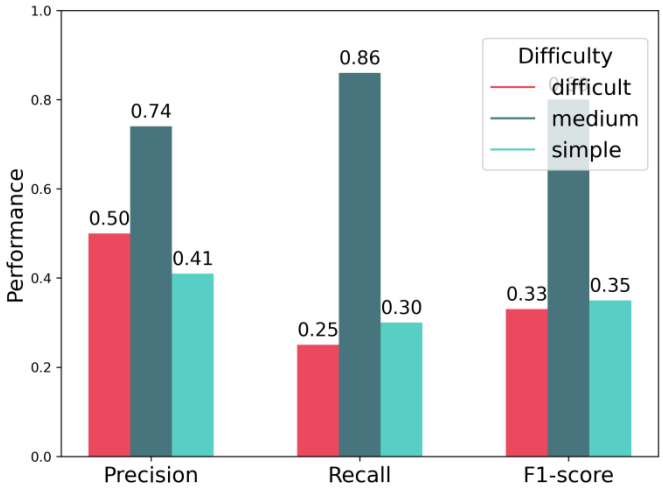

Figure 8. Evaluation results of MLP on words of different difficulty levels

Based on the given data, we used a Multilayer Perceptron (MLP) to classify word-guessing problems of different difficulty levels. After training and testing the model, we obtained the following evaluation report: our model achieved an accuracy of 67.59%, precision of 64.42%, recall of 67.59%, and an F1-score of 0.65. The model performed well overall.

Next, we can see the classification performance of each category. For problems of difficulty level "difficult", the model had a low precision of only 50% and a recall of 25%, with an F1-score of 0.33. For problems of difficulty level "medium", the model performed the best with a precision and recall of 74% and 86%, respectively, and an F1-score of 0.80. For problems of difficulty level "simple",

the model had low precision and recall, with values of 41% and 30%, respectively, and an F1-score of 0.35.

Taking all categories into consideration, we obtained the weighted average results. The overall weighted average precision was 64%, recall was 68%, and F1-score was 0.65. Although our model had some shortcomings in classifying word-guessing problems of certain difficulty levels, the MLP classifier performed well on this dataset and can be used for difficulty level classification tasks. We used the trained MLP model to classify the word "EEIER" on March 1, 2023 and found that it had a difficulty level of "medium".

References

[1] Huang, Q., Kang, Z., Zhang, Y., & Yan, D. (2020). Tool Remaining Useful Life Prediction based on Edge Data Processing and LSTM Recurrent Neural Network. In Proceedings of the 2020 IEEE International Conference on Prognostics and Health Management (ICPHM) (pp.1-5). Detroit, MI, USA. doi: 10.1109/ICPHM49022.2020.9187037.

[2] Mahajan, A., Patel, J., Parmar, M., Abrantes Joao, G. L., Shekokar, K., & Degadwala, S. (2020). 3-Layer LSTM Model for Detection of Epileptic Seizures. In 2020 Sixth International Conference on Parallel, Distributed and Grid Computing (PDGC) (pp. 447-450). Waknaghat, India: IEEE. doi: 10.1109/PDGC50313.2020.9315833.

[3] De Silva, N. "Selecting Optimum Seed Words for Wordle using Character Statistics." In 2022 Moratuwa Engineering Research Conference (MERCon), pp. 1-6. Moratuwa, Sri Lanka, 2022. doi:10.1109/MERCon55799.2022.9906176.

[4] Wang, W., Guan, J., Che, X., & Wang, W. "MS-MLP: Multi-scale Sampling MLP for ECG Classification." In 2022 30th European Signal Processing Conference (EUSIPCO), pp. 1288-1292. Belgrade, Serbia, 2022. doi: 10.23919/EUSIPCO55093.2022.9909814.

[5] Jahromi, A. H., & Taheri, M. "A non-parametric mixture of Gaussian naive Bayes classifiers based on local independent features." In 2017 Artificial Intelligence and Signal Processing Conference (AISP), pp. 209-212. Shiraz, Iran, 2017. doi: 10.1109/AISP.2017.8324083.