# A comparative study for COVID-19 cases forecasting with loss function as AIC and MSE in RNN family and ARIMA

1st Naoki Dohi
*Koch Univercity of Technology*
Kochi, Japan
255055c@gs.kochi-tech.ac.jp

2nd Namal Rathnayake
*Kochi University of Technology*
Kochi, Japan
namalhappy@gmail.com

3rd Yukinobu Hoshino
*Kochi University of Technology*
Kochi, Japan
hoshino.yukinobu@kochi-tech.ac.jp

2022 Joint 12th International Conference on Soft Computing and Intelligent Systems and 23rd International Symposium on Advanced Intelligent Systems (SCIS&ISIS) | 978-1-6654-9924-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/SCISISIS55246.2022.10001870

***Abstract*—Forecasting COVID-19 incidents is a trending research study in today's world. Since Machine learning models have been occupied in forecasting recently, this study focus on comparing statical and machine learning models such as ARIMA, RNN, LSTM, Seq2Seq, and Stacked LSTM. The performances were evaluated using two loss functions, namely, AIC and RMSE. The results showed that RNN performs with the lowest RMSE with -49.5% compared with the ARIMA. Seq2Seq scored the highest correlation of determination (R2) with 0.92.**

***Index Terms*—COVID-19, Forecasting, RNN, LSTM, Seq2Seq, Stacked LSTM, ARIMA, Statistical models, AIC, MSE, R2**

## I. Introduction

### A. Background

Over the past two years, SARS-CoV-2 has affected the entire world. According to the World Health Organization(WHO) weekly report [1], the number of confirmed cases and deaths worldwide has exceeded 510 million and 6 million, respectively. In addition, according to a dataset published by Johns Hopkins University(JHU) [2], the number of infected people has exceeded 9.41 million, and the number of deaths has exceeded 30,000. The impact of the disease has changed lifestyles and significantly affected economic activities. Accordingly, predictive models must be developed for political decision-making on public health, hospital beds, vaccines, and emergency declarations. In addition, the MonkeyPox virus has recently appeared, and WHO is warning each country [3]. Therefore, we believe that constructing predictive models for infectious diseases will continue to be important in the future.

### B. Related Work

The study by Wu et al. compares statistical and neural network models in the field of epidemic forecasting [4]. In this study, Auto-Regressive Integrated Moving Average(ARIMA) [5], Long-Short Term Memory(LSTM) [6] Sequence to Sequence with Attention, and Transformer [7], are compared in the four models. In this case, ARIMA's Root Mean Square Error (RMSE) was set as the criterion for comparison. The Centers for Disease Control and Prevention (CDC) dataset used features from the past ten weeks to predict the previous week's ratio of influenza-like illness outbreaks in the last four weeks. Their study results showed that Transfomer was able to reduce the RMSE the most relative to the RMSE of ARIMA.

Spyros et al. (2018) researched whether machine learning models such as Recurrent Neural Networks can replace statistical models of classical time series forecasting such as ARIMA [8]. A comparative study was conducted on the forecasting accuracy of statistical and machine learning models using data from the M3 competition. The results showed that statistical models tended to have more minor forecasting errors. However, they noted that the results of this study might be dataset-dependent. Therefore, when researching time series forecasting, it is necessary to compare statistical and machine learning models to construct a model with higher forecasting accuracy.

In recent years, transformer [7] has been adapted to various domains. Vision transformer [9] has been used in the computer vision domain. Moreover, in the natural language processing domain, transformer [7] has been applied to create many transformer-based models such as BERT [10].

In the area of time series forecasting, Zhou et al., in their work [11], proposed the Informer model, which is a modified version of the Transformer for time series. The Informer model often predicts in a single prediction rather than recurrently using the predictions as input for the next prediction.

### C. Purpose

In time series forecasting using machine learning, learning does not converge and often results in overlearning. This scenario was confirmed in a previous study by Spyros et al. [8], who stated that LSTM has a significant prediction error relative to model fitting. They also recommend comparing statistical and machine learning models when making time series forecasts because most of the studies on time series forecasting use only one of the two approaches, statistical modelling or machine learning, and their opinions are biased.

This study uses several neural network models (RNN, LSTM, CNN, and Seq2Seq) that make multi-time forecasts in a single forecast, with Akaike information criterion(AIC) and Root Mean Square Error(RMSE) as loss functions. Com-

parative validation was conducted using the RMSE of ARIMA as a reference for building a general-purpose model.

## II. Methodology

### A. Autoregressive Integrated Moving Average(ARIMA)

The difference series is expressed as Eq.2 with the lag operator $L^d$, which means the differences between time points, are defined as Eq.1. Thus, d-orders differences series defines as $\Delta^d y_t$(where $\Delta^0 y_t = y_t$). Furthermore, ARIMA is formulated as Eq.3 where $\phi$ is auto-correlation coefficients, $\theta$ is moving average coefficients, $\varepsilon_t$ is white noise. It is used as a representational traditional statistical model compared with machine learning.

$$L^d y_t = y_{t-d} \tag{1}$$

$$\Delta^d y_t = (1 - L^d) y_t \tag{2}$$

$$(1 - \sum_{i=1}^{p} \phi_i L^i) \Delta^d y_t = (1 + \sum_{j=1}^{q} \theta_j L^j) \varepsilon_t \tag{3}$$

### B. RNN

RNN proposed by Elman recursively learns series [12]. It formulated the hidden layer for RNN. as Eq.4. where, time is $t$, inputs is $\boldsymbol{x}_t$, hidden state vector is $\boldsymbol{h}_t$, the weights for inputs are $\boldsymbol{W}_x$, and the weights for hidden state vector are $\boldsymbol{W}_h$. It is used as a representational model for comparison with a statistical model.

$$\boldsymbol{h}_t = \tanh\left(\boldsymbol{x}_t \boldsymbol{W}_x + \boldsymbol{h}_{t-1} \boldsymbol{W}_h + \boldsymbol{b}\right) \tag{4}$$

### C. Long-Short Time Memory(LSTM)

In 1997, Hochreiter and Schmidhuber proposed LSTM [6]. This method solves the problem of rescue dependence for the long-term in RNN. LSTM block is formulated as Eq.5 and Eq.10. Where, the subscripted $\boldsymbol{W}$ in each equation are the weight for the input $\boldsymbol{x}_t$, and the subscripted $\boldsymbol{U}$ are the weight for the hidden state vector $\boldsymbol{h}_{t-1}$ at the previous time. Since the same equation has different gates, the weights and biases in each equation are subscripted with the gate for the weights.

$$\boldsymbol{f} = \sigma(\boldsymbol{x}_t \boldsymbol{W}_f + \boldsymbol{h}_{t-1} \boldsymbol{U}_f + \boldsymbol{b}_f) \tag{5}$$

$$\boldsymbol{g} = \tanh(\boldsymbol{x}_t \boldsymbol{W}_g + \boldsymbol{h}_{t-1} \boldsymbol{U}_g + \boldsymbol{b}_g) \tag{6}$$

$$\boldsymbol{i} = \sigma(\boldsymbol{x}_t \boldsymbol{W}_i + \boldsymbol{h}_{t-1} \boldsymbol{U}_i + \boldsymbol{b}_i) \tag{7}$$

$$\boldsymbol{o} = \sigma(\boldsymbol{x}_t \boldsymbol{W}_o + \boldsymbol{h}_{t-1} \boldsymbol{U}_o + \boldsymbol{b}_o) \tag{8}$$

$$\boldsymbol{c}_t = \boldsymbol{f} \odot \boldsymbol{c}_{t-1} + \boldsymbol{g} \odot \boldsymbol{i} \tag{9}$$

$$\boldsymbol{h}_t = \boldsymbol{o} \odot \tanh\left(\boldsymbol{c}_t\right) \tag{10}$$

LSTM is able to hold more long-term trends than RNN. Thus, It is used for forecasting COVID-19 and confirmed in this study.

Furthermore, It is built as a three-layer model that includes one Input layer, one recurrent layer, and one output layer and a four layers model including one Input layer, two recurrent layers, and one output layer.

### D. I/O relationship for RNN/LSTM

RNN families are able to have many relationship types for I/O. In this study, the architecture of RNN uses the Many to Many input-output relationship structures to forecast the following seven days with seven days inputs as shown in Fig.1.

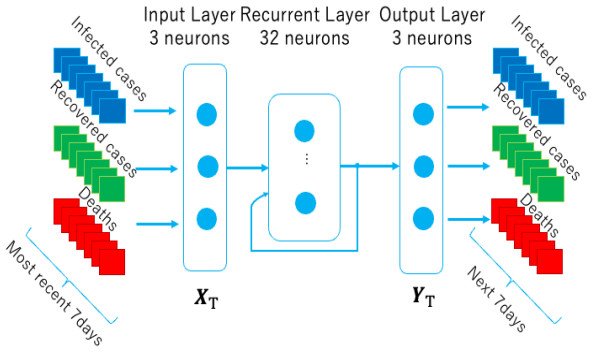


Fig. 1. Many to Many I/O

The Many To Many are considered to minimize the error with the label data at each time. Also, This allows for multiple time forecasts to be made simultaneously. In this case, Many To Many's RNN, LSTM, Stacked-LSTM, and Seq2Seq, which will be described later, are used because It is needed to forecast multiple horizons.

### E. Seq2Seq

Seq2Seq consists of Encoder and Decoder, and LSTM is used as a hidden layer in Encoder and Decoder. Fig.2 shows model architecture. In this case, Next week's data is forecast by the context vector$\boldsymbol{c}$, which is compressed from inputs one week from Encoder outputs. In this study, Seq2Seq was used to find how the use of context vectors affected the prediction and discussed how it differs from Stacked-LSTM.

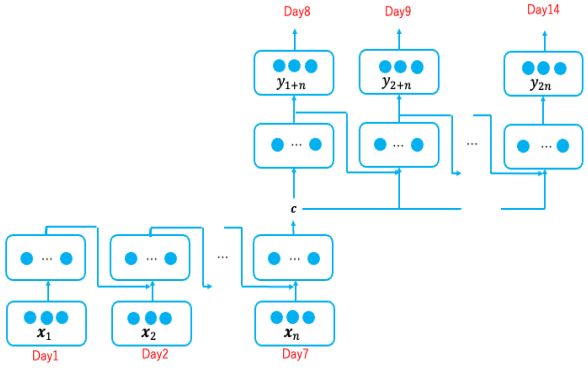


Fig. 2. Seq2Seq

### F. CNN

CNN can find Local features because it uses a filter matrix to operate convolution. Convolution is the operation with addition and multiplication between the input and filter matrix. For

time series data, The trend as a moving average is getting by striding filters. Thus, this study is used to get the time series trend depending on COVID-19.

### G. L2norm

Regularization can be achieved by adding a penalty term to the loss function to minimize the value of the new loss function $E_w$ in equation11. Models are learned to determine the weight $w$ that minimizes the value of the loss function $E$. Note that $\lambda$ is a parameter indicating the strength of regularization.

$$E_w = E + \lambda \sum_{k=1}^{n} w_k^2 \tag{11}$$

Since the size of $\lambda$ is proportional to the size of the value for the loss function, $\lambda$ is adjusted to reduce the weight. In this case, $\lambda$ was set to $0.001$ and adapted the L2 norm to RNNs, LSTMs, and CNNs, which are often over-fitting.

### H. Loss functions/Evaluate function

In this paper, The functions in sectionII-H1 and sectionII-H2 use for loss functions. In addition to these functions, The function in sectionII-H3 use for the evaluate functions.

*1) Root Mean Squere Error(RMSE):* Each models are evaluated by Root Mean Square Error(RMSE), is formulated as Eq.**??**

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{12}$$

Where, $y_i$ is observed value, $haty_i$ is predicted value, and $n$ is number of data. RMSE increases when data outliers are included

*2) Akaike Information Criterion(AIC):* Akaike Information Criterion(AIC) is proposed by Akaike [13] [14]. An evaluation model fitting with the observed value. The smaller the AIC value is, The better model fits. AIC is formulated as Eq.13 and Eq.14. Where $k$ is the number of parameters in the model, $L$ is the likelihood, $N$ is the number of samples, and RSS is short for residual square error.

$$\begin{aligned} AIC &= 2k - 2\ln(\hat{L}) && (13)\\ &= 2k + N\ln 2\pi + N\ln\frac{RSS}{N} + N && (14)\end{aligned}$$

In fact, It is calculated by the formula15 because $N$ and $\ln 2\pi$ are constants.

$$AIC = N\ln\frac{RSS}{N} + 2k \tag{15}$$

Note that when used as a loss function, it was defined as Eq.16 in the implementation.

$$AIC = \ln\frac{RSS}{N} + 2k \tag{16}$$

In this time, AIC is used as indexes for building model considered long-term trend.

*3) R2:* R2 is an index of the model for fitting the observed values. It is formulated as Eq.17. Note that $Y$ is the observed value, $\hat{y}$ is the predicted value, and $\bar{y}$ is the average of the observed values. It is used as an index for model fitting in the same way as AIC.

$$R2 = 1 - \frac{\sum(y-\hat{y})^2}{\sum(y-\bar{y})^2} \tag{17}$$

## III. Experiment

### A. Experiment Order

The experiment is flowed by order below.

1) Data collection
2) Building model for baseline(ARIMA)
3) Scaling data and creating data set for deep learning
4) Building deep learning model
5) Evaluation function for evaluation and comparison of model

In this study, data sets supplied by Johns Hopkins University (JHU) in the USA are used [2]. This dataset comprises accumulated data for each country, including confirmed cases, recovered cases, and death. After downloading these data sets, new data sets are constructed. New data sets are constructed by operation like $y_t - y_{t-1}$. Because data sets do not include new cases every day. This study used data from 28/10/20 to 23/01/20, composed of train data (from 23/01/20 to 21/10/20) and test data (from 22/10/20 to 28/10/20).

The next step is building a model for baseline(ARIMA)

ARIMA has two orders: the order of how many correlations with past time are included in the model at time$t$, and how many differencing are taken. Usually, the order of AR is defined as $p$, the order of MA as $q$, and the order of I as $d$. ARIMA is usually challenging to determine the order by ACF or PACF due to the model's flexibility. In this study, $ARIMA(2,1,3)$ was selected from the AIC by searching for the order in the range of I in the following table.

TABLE I
THE ORDER RANGE OF $(p,d,q)$

| | p | d | q |
|---|---|---|---|
| min | 1 | 0 | 0 |
| max | 3 | 1 | 3 |

Next, the preprocessing for deep learning is executed. At first, the scaling was performed. scaring is formulated by Eq.18 and Eq.19.

$$\begin{aligned} x_{std} &= \frac{x - x_{min}}{x_{max} - x_{min}} && (18)\\ x_{scaled} &= x_{std}(max - min) + min && (19)\end{aligned}$$

Note that $x_{min}$ are zero at all variables, $x_{max}$ are 1762, 3941, and 29 in infected cases, recovered cases, and death, respectively. The scaling range is minus one to plus one. Second, create train and test data sets. Data sets are built, such as train data for a week with test data for next week.

TABLE II
EVALUATION VALUE OF EACH MODEL IN THE TRAINING TERM

| | CNN | RNN | LSTM | Seq2Seq | Stacked |
|---|---|---|---|---|---|
| **RMSE** | 161 | 128 | 128 | 107 | 118 |
| **AIC** | 2786 | 5082 | 11997 | 28539 | 28592 |
| **RMSE** | 179 | 149 | 206 | 135 | 173 |
| **AIC** | 2845 | 5163 | 12249 | 28664 | 28797 |

Next, deep learning models such as CNN, RNN, LSTM, Seq2Seq, and Stacked LSTM were built.

The input values were vector variables for seven days, including the number of new infections, new recoveries and new deaths. The output values were vector variables, including the number of new infections, new recoveries and new deaths for the following week.

For CNN settings, the number of filters and kernels was taken as three, the stride was 1, and the padding was causal. The model was built for RNN and LSTM settings with three neurons in the input layer, three in the output layer, and 32 in the hidden layer.

For stacked LSTM settings, the hidden layer of the above LSTM was made into two layers. The number of neurons was set to 32 in both layers. Similarly, Seq2Seq had one LSTM layer for Encoderand, one LSTM layer and one output layer for Decoder, with the number of neurons in the LSTM layer set to 32 and the number of neurons in the output layer set to 3. The L2 norm was not used for Seq2Seq and Stacked LSTM. All models were trained with online learning and Adam as the optimization algorithm.

Finally, these models are evaluated with AIC, RMSE, and R2. Additionally, these were compared fluctuation ratio of error with ARIMA as the baseline in RMSE.

### *B. Result*

Table II shows the evaluation values for each model in the training term. Here, the upper part shows the models with MSE as the loss function and the under part shows the models with AIC as one. From this table, the model that minimized RMSE was Seq2Seq with MSE. This can be said that it is better to use MSE as one for small periodic fluctuations. Furthermore, taking the average of the RMSE for each loss function, MSE is 128.4, and AIC is 168.4, indicating that the RMSE tends to be smaller when the loss function is the MSE. The average value of AIC was 13239.2 for MSE loss one and 15541.6 for the AIC loss one. This shows that AIC also tends to be minimized when MSE is used as the loss one.

Table III shows the AIC and the prediction accuracy for ARIMA over the test term. Based on these results, we discuss them in the sectionIV for the constructed model.

TABLE III
AIC AND PREDICTION ERROR OF ARIMA FOR TEST TERM

| | AIC | RMSE |
|---|---|---|
| **ARIMA** | 3361 | 103 |

TABLE IV
VALUES EVALUATED FOR EACH MODEL DURING THE TEST PERIOD

| | CNN | RNN | LSTM | Seq2Seq | Stacked |
|---|---|---|---|---|---|
| **RMSE** | 118 | 52 | 78 | 120 | 113 |
| **Gap(%)** | 14.6 | -49.5 | -24.3 | 16.5 | 9.7 |
| **RMSE** | 143 | 70 | 197 | 118 | 102 |
| **Gap(%)** | 38.8 | -32.0 | 91.2 | 14.6 | -0.9 |

TABLE VI
R2 FOR EACH MODEL OVER THE ENTIRE PERIOD

| | CNN | RNN | LSTM | Seq2Seq | Stacked |
|---|---|---|---|---|---|
| **R2** | 0.82 | 0.89 | 0.88 | 0.92 | 0.90 |
| **R2** | 0.77 | 0.84 | 0.70 | 0.87 | 0.79 |

Table IV shows the evaluation values for each model in the test term. Here, The upper part shows the models in settings on MSE as the loss function, and the under part shows the models in settings on AIC as the loss one.

In the MSE settings, RNN minimized prediction error, RMSE, most concerning ARIMA, has RMSE=103. Furthermore, in the AIC settings, RNN minimized RMSE.

Table V shows the ratio of increase/decrease in Wu et al.'s influenza-like illness forecasting model [4] concerning ARIMA (where Seq2Seq is with Attention). The table shows that the RNN for COVID-19 forecasting with MSE as the loss function reduced the error by 7.1% more than the transformer for influenza-like illness forecasting with RMSE for ARIMA as the reference. These results differ from those described in the Makridakis et al. study. [8].

TABLE V
EVALUATION OF THE INFLUENZA LIKE ILLNESS FORECASTING MODEL [4] BY WU ET AL.

| | ARIMA | LSTM | Seq2Seq | Transformer |
|---|---|---|---|---|
| **Gap** | 0.0 | -20.9 | -37.1 | -42.4 |

Table VI shows the results of calculating R2 for the forecasts for all periods. These results show that Seq2Seq and Stacked LSTM with MSE as the loss one have an R2 above 0.90, indicating a high degree of model adaptation. RNN and LSTM are also highly adaptive, Because all the facts of R2 close to 0.90.

## IV. DISCUSSION

In Table.II, both RMSE and AIC were smaller when the loss function was set to MSE rather than AIC. This is because the second term of AIC should have functioned as a limit on the increase in the number of model parameters when calculating the loss function.

However, it did not function well because the second term, which is a constant, became zero due to differentiation during backpropagation. In order to use it as a loss function, it is necessary to devise a way to limit the increase in the number

of parameters in the model while at the same time ensuring that the value of the derivative does not become zero.

Table IV shows that the RMSE over the test period was more significant for CNN, RNN, and LSTM models with AIC in the loss function.

On the other hand, for Seq2Seq and Stacked LSTM, the RMSE was smaller for the models with AIC in the loss function, suggesting that AIC penalizes the robustness of the model. Therefore, the RMSE was more significant for the single-layer models because the models whose predictions were evaluated to be close to the moving average were evaluated.

This is considered to be the reason why the RMSE was more extensive in the single-layer model. On the other hand, in the Stacked LSTM and Seq2Seq models, the model became more complex due to the layering of layers. While learning the trend component, it was also possible to learn small amplitudes, which is why the AIC was suitable even for small models.

From Table.VI. Model fitting of the Stacked LSTM and Seq2Seq was high. This can be attributed to the multi-layered model being more flexible and capturing long-term trends. This can also be confirmed by the LSTM results shown in Figure3 and the Seq2Seq results shown in Figure4.

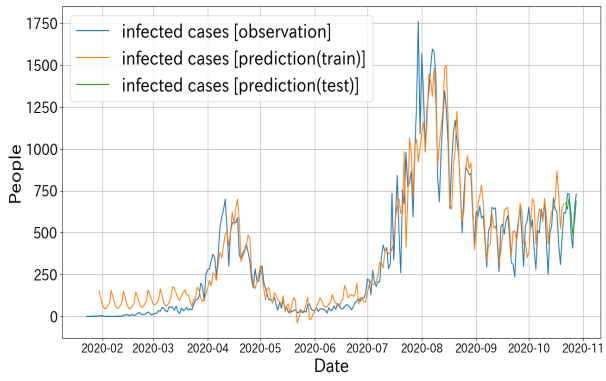


Fig. 3. The result of LSTM

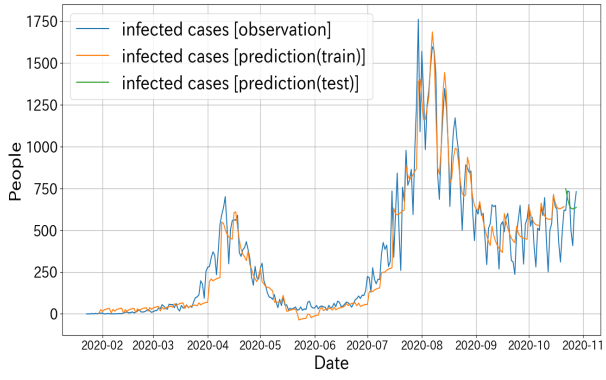


Fig. 4. The result of Seq2Seq

Seq2Seq and the Stacked LSTM were a good fit, but the RMSE of the prediction accuracy of the test interval was more significant than that of the other models. This may be because the models could detect the trend strongly, so they did not react robustly to the period with a short test period but only to the upward trend of the test period.

Finally, as can be seen by comparing Figures 3 and 4, LSTM has a bias in the forecast before the first wave starts and before the third wave starts. On the other hand, Seq2Seq could predict the long-term trend properly without any additional bias. We believe this is because Seq2Seq uses a context vector, a multi-layered, one-week summary of infection status, as input to the decoder.

## V. Conclusion

In this study, the models were built with two loss functions: MSE and AIC. As a result, the RMSE of RNN was the smallest, with a -49.5% decrease from ARIMA, and the R2 of Seq2Seq was the highest, with 0.92. In the future, we will optimize the implementation of the loss function (AIC) for medium- and long-term trend prediction and compare it with other infectious disease data models.

## References


[1] World Health Organization. "COVID-19 weekly epidemiological update, edition 91, 11 May 2022," World Health Organization, 2022

[2] https://github.com/CSSEGISandData/COVID-19

[3] World Health Organization. "Risk communication and community engagement (RCCE) for monkeypox outbreaks: Interim guidance, 24 June 2022," World Health Organization, 2022

[4] Wu, Neo and Green, Bradley and Ben, Xue and O'Banion, Shawn. "Deep transformer models for time series forecasting: The influenza prevalence case," arXiv preprint arXiv:2001.08317, 2020.

[5] Box, George EP and Jenkins, Gwilym M and Reinsel, Gregory C and Ljung, Greta M. "Time series analysis: forecasting and control," John Wiley & Sons, 2015

[6] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory," Neural computation, Vol.9, No.8, pp.1735–1780, 1997.

[7] Vaswani, Ashish and Shazeer, Noam and Parmar, Niki and Uszkoreit, Jakob and Jones, Llion and Gomez, Aidan N and Kaiser, Łukasz and Polosukhin, Illia. "Attention is all you need," Advances in neural information processing systems, Vol.30, 2017.

[8] Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. "Statistical and machine learning forecasting methods: Concerns and ways forward," PloS one, Vol.13, No.3, pp.e0194889, 2018.

[9] Dosovitskiy, Alexey and Beyer, Lucas and Kolesnikov, Alexander and Weissenborn, Dirk and Zhai, Xiaohua and Unterthiner, Thomas and Dehghani, Mostafa and Minderer, Matthias and Heigold, Georg and Gelly, Sylvain and others. "An image is worth 16x16 words: Transformers for image recognition at scale," arXiv preprint arXiv:2010.11929, 2020

[10] Devlin, Jacob and Chang, Ming-Wei and Lee, Kenton and Toutanova, Kristina. "Bert: Pre-training of deep bidirectional transformers for language understanding," arXiv preprint, arXiv:1810.04805, 2018.

[11] Zhou, Haoyi and Zhang, Shanghang and Peng, Jieqi and Zhang, Shuai and Li, Jianxin and Xiong, Hui and Zhang, Wancai. "Informer: Beyond efficient transformer for long sequence time-series forecasting," Proceedings of AAAI, 2021

[12] Jeffrey L Elman. "Finding structure in time," Cognitive science, Vol.14, No.2, pp.179–211, 1990.

[13] Hirotugu Akaike. "A new look at the statistical model identification," IEEE transactions on automatic control, Ieee, Vol.19, No.6, p.716-p.723, 1974.

[14] Hirotogu Akaike. "Information theory and an extension of the maximum likelihood principle," Springer, p.199-p.213,1998.