EVEREST ENGINEERING COLLEGE

(Affiliated to Pokhara University)
Sanepa-2, Lalitpur

[Subject Code:CMP390]
A MINOR PROJECT PROPOSAL ON
“

Analyzing Univariate Time Series Machine Learning Algorithms/Models With Different Time Gap And Forecasting Average Surface Temperature of Kathmandu Valley

”

SUBMITTED BY

PRABIN BAHADUR THAPA	**[19120071]**
PRABIN BHUSAL	**[19120072]**
PRASANNA ADHIKARI	**[19120075]**

SUBMITTED TO
DEPARTMENT OF INFORMATION TECHNOLOGY ENGINEERING
EVEREST ENGINEERING COLLEGE
SANEPA, LALITPUR
September, 2022

COPYRIGHT

The author has given permission for this report to be made freely available for view by anybody at Everest Engineering College's library. Additionally, the author has consented that the lecturers who oversaw the project work documented here, or, in their absence, the head of the department where the project report was completed, may grant permission for widespread copying of this project report for scholarly purposes. It is agreed that credit will be provided to the project report's author and the Department of IT and Computer, EEC, for any usage of the project report's content. This report may not be copied, published, or used in any other way for commercial advantage without the Department's and the author's explicit consent. Requesting authorization to use or copy of the material in this report in whole or in part should be addressed to:

Head
Department of IT Engineering
Everest Engineering College
Sanepa, Lalitpur

CERTIFICATE

The undersigned attests to having read and recommended for approval to the Department of Information Technology and Computer Engineering a project report titled "**Analyzing Univariate Time Series Machine Learning Algorithms/Models With Different Time Gap And Forecasting Average Surface Temperature of Kathmandu Valley**" that was submitted by Prabin Bahadur Thapa, Prabin Bhusal and Prasanna Adhikari in partial fulfillment of the requirement for the Bachelor's degree in Information Technology Engineering.

……………………………..

Er. Anuj Ghimire

Head

Department of IT Engineering

Everest Engineering College,

Sanepa, Lalitpur

ACCEPTANCE

The project report titled "**Analyzing Univariate Time Series Machine Learning Algorithms/Models With Different Time Gap And Forecasting Average Surface Temperature of Kathmandu Valley**" which was turned in by Prabin Bahadur Thapa, Prabin Bhusal and Prasanna Adhikari in partial fulfillment of the requirement for the Bachelor's degree in Information Technology Engineering, has been accepted as a true record of the work independently completed by the group in the department.

……………………………..

Er. Birod Rijal

Principal

Everest Engineering College,

Sanepa, Lalitpur

……………………………..

Er. Anuj Ghimire

Head

Department of IT Engineering

Everest Engineering College,

Sanepa, Lalitpur

ACKNOWLEDGEMENT

We want to express our special gratitude to Dr. Shailesh B. Pandey, Nischal Regmi for their assistance in putting the parts together and project guidance. Last but not least, we would like to express our profound gratitude to the EEC College administration, head of departments, all teaching and non-teaching staff members, and friends who helped to complete this project either directly or indirectly.

We would highly appreciate and heartily welcome the suggestions for further improvement if any.

-Prabin Bahadur Thapa

-Prabin Bhusal

-Prasanna Adhikari

ABSTRACT

The time-series forecasting field is crucial and encourages ongoing research into areas of interest for various applications. Choosing the appropriate number of historical observations is an important step in time-series forecasting (lags). This project investigates forecasting accuracy based on the selection of an appropriate time-lag value between Regression method and LSTM (Long-Short Term Memory) in stock price prediction.The performance metrics were: Root Mean Square Error (RMSE), and R-squared. The investigation demonstrated that the both proposed algorithms were able to provide high accuracy with least RMSE and R-squared for respective time gaps and forecasting period.

Keywords: Recurrent Neural Network; time-series forecasting;average surface temperature; analysis; prediction; statistics; machine learning;ARIMA; LSTM;

Table of content

List of Figures

List of Tables

Chapter 1: Introduction

Univariate time series models are a class of specifications where one attempts to model and to predict financial variables using only information contained in their own past values and possibly current and past values of an error term. So basically, in an univariate model, only one variable is looked at for any of its forecasting.

Time series forecasting is a technique for the prediction of events through a sequence of time. The technique is used across many fields of study, from geology to behavior to economics. The techniques predict future events by analyzing the trends of the past, on the assumption that future trends will hold similar to historical trends. Long Short-Term Memory(LSTM) and ARIMA are two widely used models for time series forecasting.

Forecasting the average surface temperature of Kathmandu valley means that we will be known with the temperature of the next day, or even later than that before the day even begins, obviously, in variance with accuracy. Forecasting the average surface temperature using machine learning algorithms helps you to predict/discover the future temperature based on the past data that has been trained in it.

ARIMA model is one of the approaches to time series forecasting. ARIMA is a statistical analysis model that uses time series data to either better understand the data set or to predict future trends. It is an autoregressive model as it is a statistical model that predicts the future value based on the past data that has been provided to it. ARIMA models aim to describe the autocorrelations in the data.

ARIMA, short for 'Auto Regressive Integrated Moving Average' is actually a class of models that explains a given time series based on its own past values, that is, its own lags and lagged forecast errors, so that the equation can be used to forecast future values.

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network capable of learning order dependence in sequence prediction problems. This is a behavior required in complex problem domains like machine translation, speech recognition and more.

LSTMs are a complex area of deep learning. It can be hard to get your hands around what LSTMs are, and how terms like bidirectional and sequence-to-sequence relate to the field.

Chapter 2: Literature Review

In 2017, K. Goswami, A. N Patowary and J. Hazarika applied the Arima model to predict monthly temperature in the Dibrugarh region in order to research on climate change and the rising of temperature. From their research and analysis, they concluded that the Arima model is adequately fitted for the historical data and that the model can add some benefit in weather forecasting. [1]

A time series is a sequence of observations recorded over a certain period of time. A simple example of time series is how we come across different temperature changes day by day or in a month. Time Series forecasting in simple words means to forecast or to predict the future value(eg-stock price) over a period of time. Time series forecasting depends on different terms like seasonality, trends and unexpected events. It takes a stationary time series. [2]

Time series forecasting is a difficult problem with no easy solution. There are countless statistical models that claim to be the best of all, yet it's never clear which model is better.
That being said, the Arima model is often a good model to start with. The Arima word is divided into three parts, AR, I, and MA. AR is an Autoregressive component, MA is defined as Moving Average and I is for integration. [3]

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network capable of learning order dependence in sequence prediction problems.This is a behavior required in complex problem domains like machine translation, speech recognition, and more.LSTMs are a complex area of deep learning. It can be hard to get your hands around what LSTMs are, and how terms like bidirectional and sequence-to-sequence relate to the field. [4]

A data-driven approach for obtaining near-term (2–20 years) regional temperature and precipitation projections utilizing local historical observations was established in this study to facilitate civil and environmental engineering applications. Given the unique characteristics of temporal correlation and skewness exhibited in individual time series of temperature and precipitation variables, a statistical time series forecasting technique was developed based on the autoregressive integrated moving average (ARIMA) model. Annual projections obtained from the ARIMA model—depending on individual series—can be interpreted as an integration of the most recent observations and the long-term historical trend.[5]

Chapter 3: Methodology

3.1 Agile Model

The model that we have used to develop this project is the Agile Model. It is one of the most popular models that is used in present days not only in Software Development but also in most of the operations that is carried out in an organization. “Agile process model” refers to a software development approach based on iterative development. Agile is the ability to create and respond to change. The authors of the Agile Manifesto choose “Agile” as the label for this whole idea because that word represented the adaptiveness and response to change which was important to their approach [8]. Agile is based on 12 principles. Agile methods break tasks into smaller interactions, or parts do not directly involve long term planning. The division of the entire project into smaller parts helps to minimize the project risk and to reduce the overall project delivery time requirements. Plans regarding the number of interactions, the duration and the scope of each iteration are clearly defined in advance. Each iteration involves a team working through a full software development life cycle including planning, requirements analysis, design, coding, and testing before a working product is shown to the client.

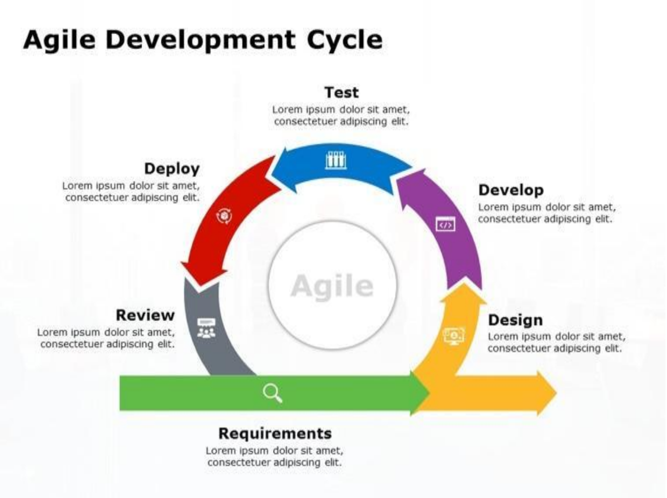

Agile Development Cycle
Test
Lorem ipsum dolor sit amet, consectetuer adipiscing elit.
Deploy
Lorem ipsum dolor sit amet, consectetuer adipiscing elit.
Develop
Lorem ipsum dolor sit amet, consectetuer adipiscing elit.
Agile
Review
Lorem ipsum dolor sit amet, consectetuer adipiscing elit.
Design
Lorem ipsum dolor sit amet, consectetuer adipiscing elit.
Requirements
Lorem ipsum dolor sit amet, consectetuer adipiscing elit.

Figure 3.1: Agile Development Cycle [7]

3.1.1 **Planning Phase**

The planning phase discussed a reason for selected goals including a detailed overview of the goals. In this phase, the first goal of the project is described to predict surface temperature. The title of this project has been selected,**”Analyzing Univariate Time Series Machine Learning Algorithms/Models With Different Time Gap And Forecasting Average Surface**

Temperature of Kathmandu Valley”. The abstract was done with all the information gathered. Then, the entire requirement that is involved in the system will be found.

3.1.2 Analysis Phase

In the analysis phase, the requirements were studied and brainstorming sessions were done to compare suitable algorithms for prediction of time series data with time gaps. Once the analysis was done a proper strategy was planned and finalized.

3.1.3 Engineering Phase

This phase involves code, test cases, results, test summary, and reports.

3.1.4 System Design

The following system design is proposed in this project .

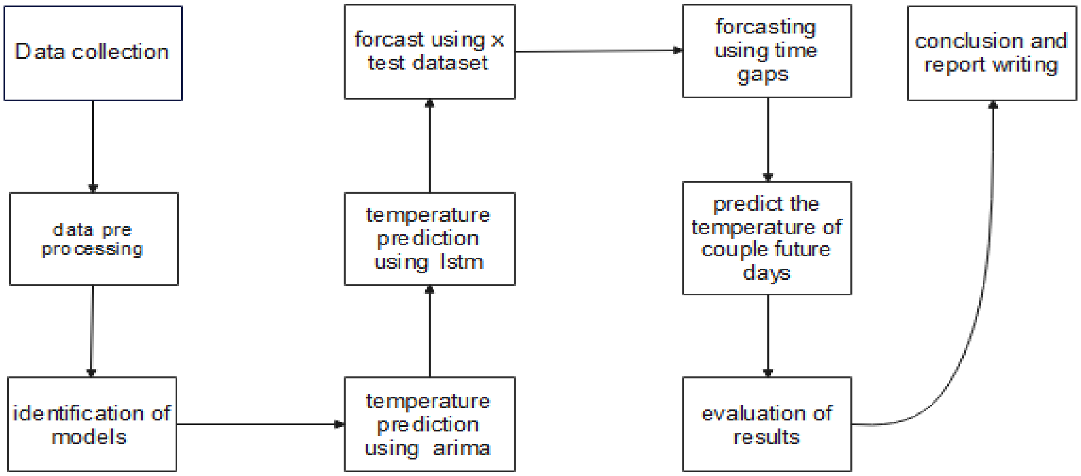

Data collection
data pre processing
identification of models
temperature prediction using arima
temperature prediction using lstm
forcast using x test dataset
forcasting using time gaps
predict the temperature of couple future days
evaluation of results
conclusion and report writing

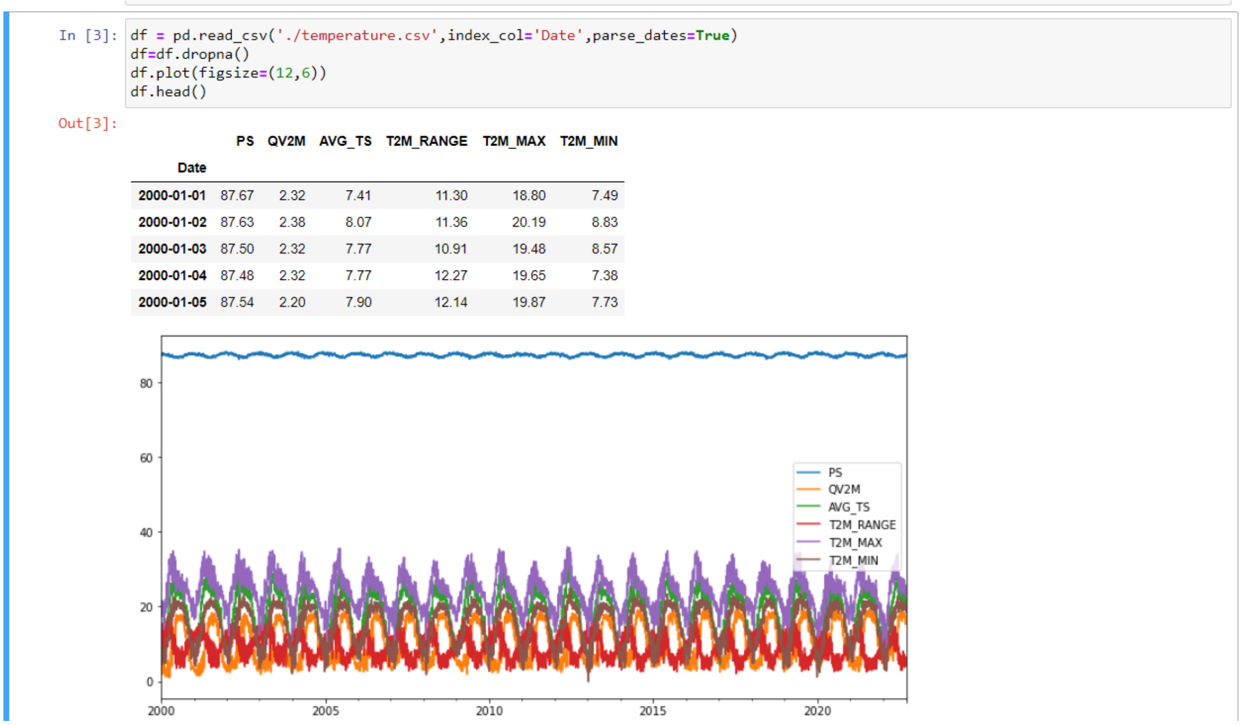
Fig 3.1.4: block diagram of proposed system

This design is logically divided into three parts. The first block represents phase 1, the second block represents phase 2, and the third & fourth block represents phase 3.
The first phase’s is a basic phase as it consists all the data collection and preprocessing parts
Phase 2 is the phase where all the models are made and forecast the temperature using the X test dataset, which was created by dividing our identical data into test and training sets.
Phase 3 involves all the testing and report writing part.

Each building block of the design is explained in the sections that follow.

3.1.4.1 Data collection

Data collection is a most crucial part and the data for the project has been taken out from nasa's website. We took the data of Kathmandu valley's average surface temperature . Then the graph is made to understand the data that we scraped from the website .



```
In [3]: df = pd.read_csv('./temperature.csv',index_col='Date',parse_dates=True)
        df=df.dropna()
        df.plot(figsize=(12,6))
        df.head()
```

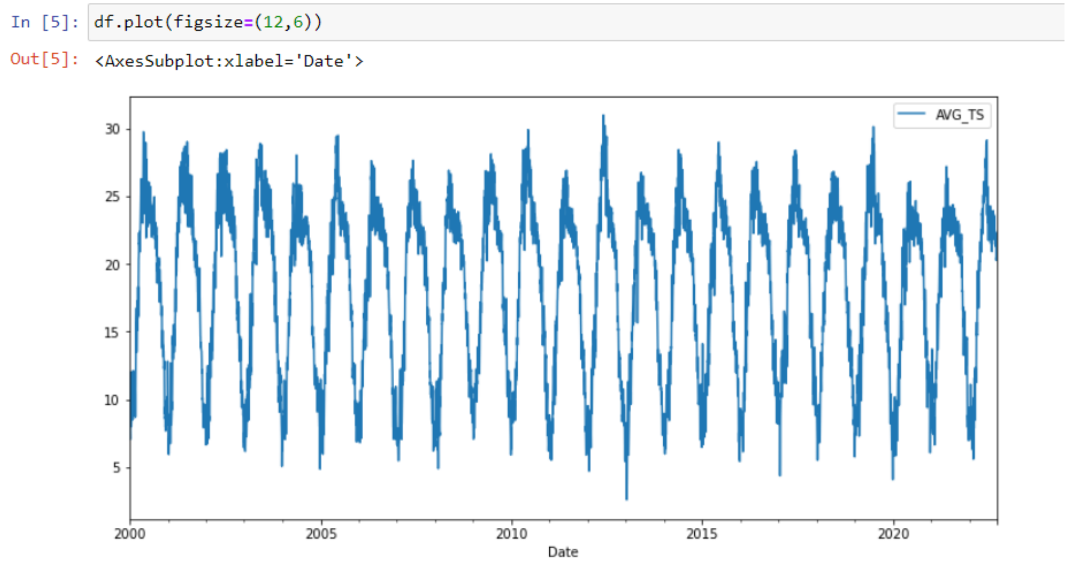
Out[3]:

	PS	QV2M	AVG_TS	T2M_RANGE	T2M_MAX	T2M_MIN
Date						
2000-01-01	87.67	2.32	7.41	11.30	18.80	7.49
2000-01-02	87.63	2.38	8.07	11.36	20.19	8.83
2000-01-03	87.50	2.32	7.77	10.91	19.48	8.57
2000-01-04	87.48	2.32	7.77	12.27	19.65	7.38
2000-01-05	87.54	2.20	7.90	12.14	19.87	7.73

Fig 3.1.4.1 : graph of our data set

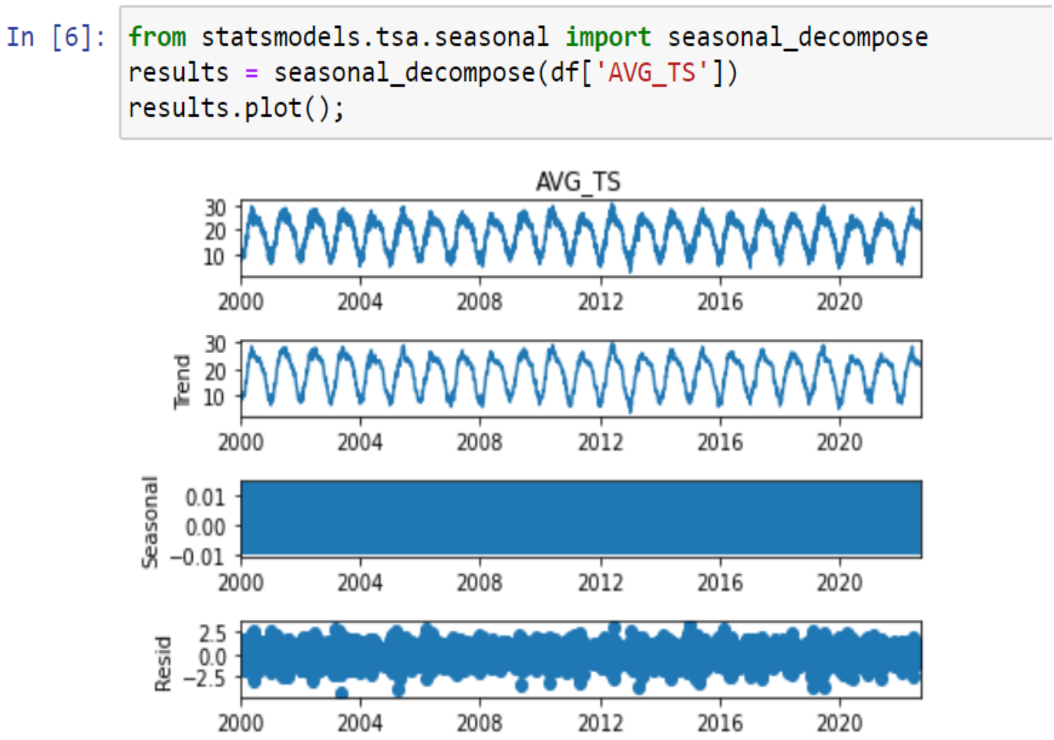
3.1.4.2 Data pre processing

After collection of data we had a lot of unnecessary data which was eliminated as part of pre-processing , then we made the artificial time gap using code which basically eliminated certain data for further use .



```
In [5]: df.plot(figsize=(12,6))
Out[5]: <AxesSubplot:xlabel='Date'>
```

Fig 3.1.4.2 : graph of average surface temperature of kathmandu from 2000 to 2022



```
In [6]: from statsmodels.tsa.seasonal import seasonal_decompose
        results = seasonal_decompose(df['AVG_TS'])
        results.plot();
```

Fig 3.1.4.3 : graph of seasonality of data

3.1.4.3 Prediction using ARIMA

An ARIMA model is a class of statistical models for analyzing and forecasting time series data.
It explicitly caters to a suite of standard structures in time series data, and as such provides a simple yet powerful method for making skillful time series forecasts.

ARIMA is an acronym that stands for AutoRegressive Integrated Moving Average. It is a generalization of the simpler AutoRegressive Moving Average and adds the notion of integration.

This acronym is descriptive, capturing the key aspects of the model itself. Briefly, they are:

- **AR**: *Autoregression*. A model that uses the dependent relationship between an observation and some number of lagged observations.
- **I**: *Integrated*. The use of differencing of raw observations (e.g. subtracting an observation from an observation at the previous time step) in order to make the time series stationary.
- **MA**: *Moving Average*. A model that uses the dependency between an observation and a residual error from a moving average model applied to lagged observations.

Each of these components are explicitly specified in the model as a parameter. A standard notation is used of ARIMA(p,d,q) where the parameters are substituted with integer values to quickly indicate the specific ARIMA model being used.

The parameters of the ARIMA model are defined as follows:

- **p**: The number of lag observations included in the model, also called the lag order.
- **d**: The number of times that the raw observations are different, also called the degree of differencing.
- **q**: The size of the moving average window, also called the order of moving average.

A linear regression model is constructed including the specified number and type of terms, and the data is prepared by a degree of differencing in order to make it stationary, i.e. to remove trend and seasonal structures that negatively affect the regression model.A value of 0 can be used for a parameter, which indicates to not use that element of the model. This way, the ARIMA model can be configured to perform the function of an ARMA model, and even a simple AR, I, or MA model.Adopting an ARIMA model for a time series assumes that the underlying process that generated the observations is an ARIMA process. This may seem obvious, but helps to motivate the need to confirm the assumptions of the model in the raw observations and in the residual errors of forecasts from the model.

3.1.4.4 Prediction using LSTM

Long Short-Term Memory (LSTM) networks are a modified version of recurrent neural networks, which makes it easier to remember past data in memory. The vanishing gradient problem of RNN is resolved here. LSTM is well-suited to classify, process and predict time series given time lags of unknown duration. It trains the model by using back-propagation. In an LSTM network, three gates are present:

a) Input Gate: Discover which value from input should be used to modify the memory. The Sigmoid function decides which values to let through 0,1. and tanh function gives weightage to the values which are passed, deciding their level of importance ranging from-1 to 1.

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

Figure 3.8: Formula of LSTM Input Gate

Here,

- Xt: Input at the current timestamp t
- bi: weight matrix of input
- ht-1: A hidden state at the previous timestamp
- Wi: Weight matrix of input associated with hidden state

Just like a simple RNN, an LSTM also has a hidden state where h(t-1) represents the hidden state of the previous timestamp and h_t is the hidden state of the current timestamp. In addition to that, LSTM also has a cell state represented by C(t-1) and C(t) for previous and current timestamps respectively.

b) <u>Forget Gate</u>: Discover what details are to be discarded from the block. It is decided by the sigmoid function. it looks at the previous state**(**ht-1**)** and the content input (Xt) and outputs a number between **0(**_omit this_) and 1(_keep this_**)** for each number in the cell state Ct−1.

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

Figure 3.9: Formula of LSTM Forget Gate

Here,

- Xt: input to the current timestamp.
- bf: weight associated with the input
- ht-1: The hidden state of the previous timestamp
- Wf: It is the weight matrix associated with hidden state

c) <u>Output Gate</u>: The input and the memory of the block is used to decide the output. Sigmoid function decides which values to let through 0,1. and tanh function gives weightage to the values which are passed deciding their level of importance ranging from-1 to 1 and multiplied with output of Sigmoid.

$$o_t = \sigma\left(W_o \ [\, h_{t-1}, x_t] \ + \ b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

Figure 3.10: Formula of LSTM Output Gate

In order to predict future trends, RNN's primary principle is to use the sequential observations learned from earlier phases. RNN has been modified with the Long-Short Term Memory (LSTM) model. The RNN's inability to detect long-term impacts can be overcome by this method.

The method of using components of previous sequences to predict future data is referred to as recurrent. The Long Short-Term Memory (LSTM) based on a "memory line" proved to be highly helpful in forecasting scenarios with long-time data because RNN cannot store long-time memory. An LSTM contains gates along the memory line that can be used to memorize previous stages.

Chapter 4: Implementation Details

4.1 Software Requirement

The list of software that will be used to run this system is listed below:

- Any operating system (Linux, Windows, MacOS)
- Web Browser
- Code Editor (Vs Code, Sublime, Atom, Jupyter Notebook, Colab)

4.2 Hardware Requirement

The list of hardware that will be used to run this system is listed below:

- General PC (min RAM 8GB, HDD 500GB, SSD adds value)

Chapter 5: Results and Analysis

We adopted two of the most commonly applied models in time series prediction to establish and compare forecasting models for monthly, weekly and daily incidence of average surface temperature of Kathmandu valley . The results demonstrated that both ARIMA and LSTM could be used to build prediction models for the average surface temperature while different models might be suitable for average surface temperature prediction at different time scales.

One of the highlights of our study is that we forecasted the average surface temperature of Kathmandu valley using the prediction models with “Rolling Forecasting Origin”, also called “walk-forward model validation”, which forecasts the incidence value by adding to the previously observed real incidence data solving the problem of connection actuality in the prediction phase of the method and increasing the accuracy of the prediction. In addition, we built models and compared the forecasting performances with three time scales including monthly, weekly and daily incidence. The results indicated that time scales should be taken into account when selecting prediction models of average surface temperature because different models might be appropriate for temperature forecasting at different time scales.

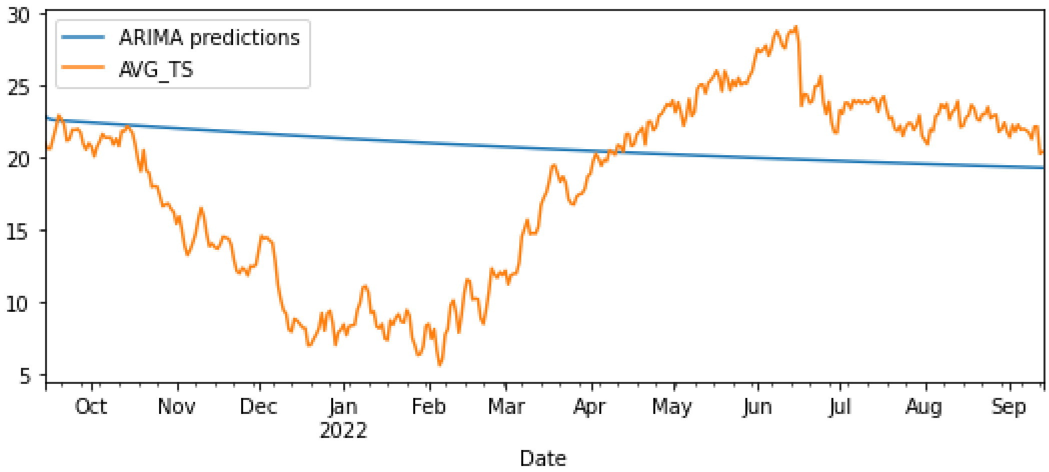


Fig 5.1: ARIMA Prediction without time gap

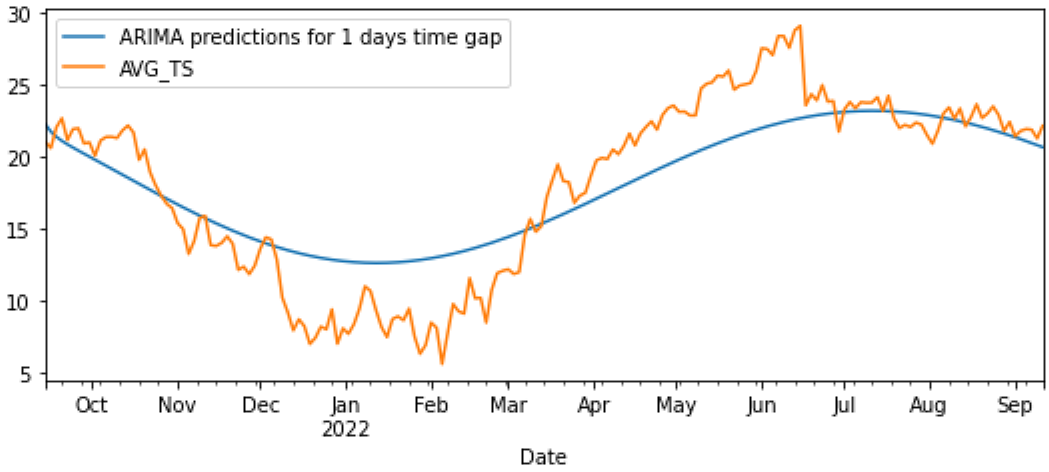


Fig 5.2: ARIMA Prediction with 1 day time gap

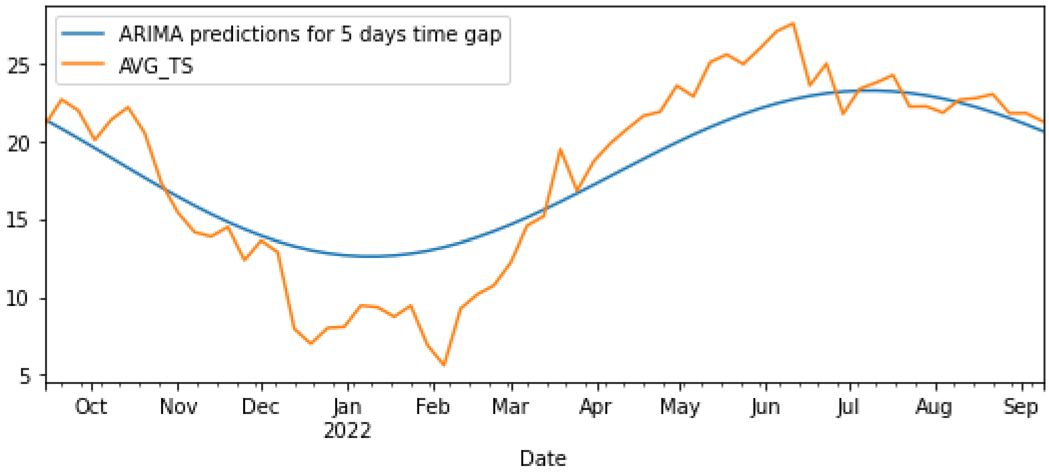


Fig 5.3: ARIMA Prediction with 5 day time gap

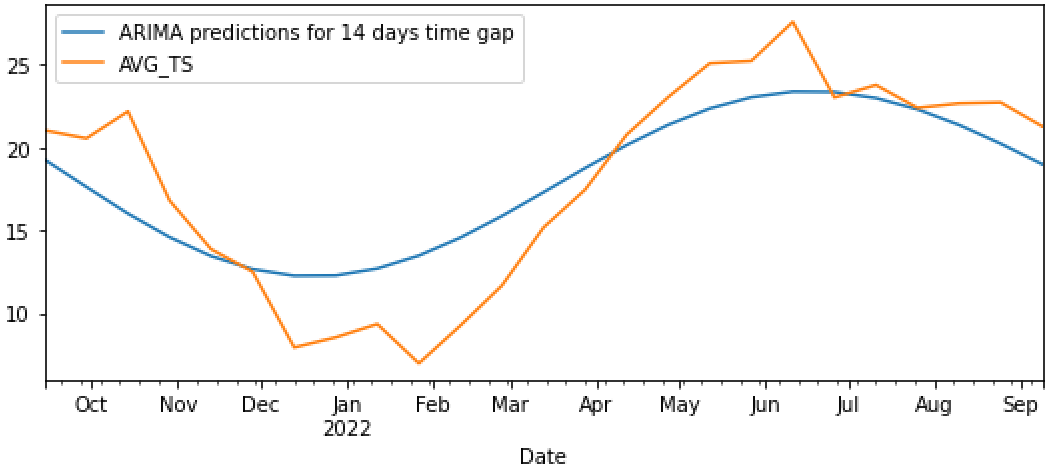
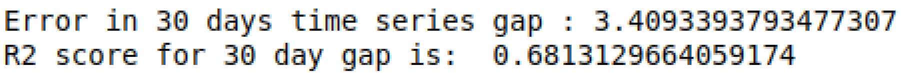


Fig 5.4: ARIMA Prediction with 14 day time gap



```
Error in 30 days time series gap : 3.4093393793477307
R2 score for 30 day gap is:  0.6813129664059174
```

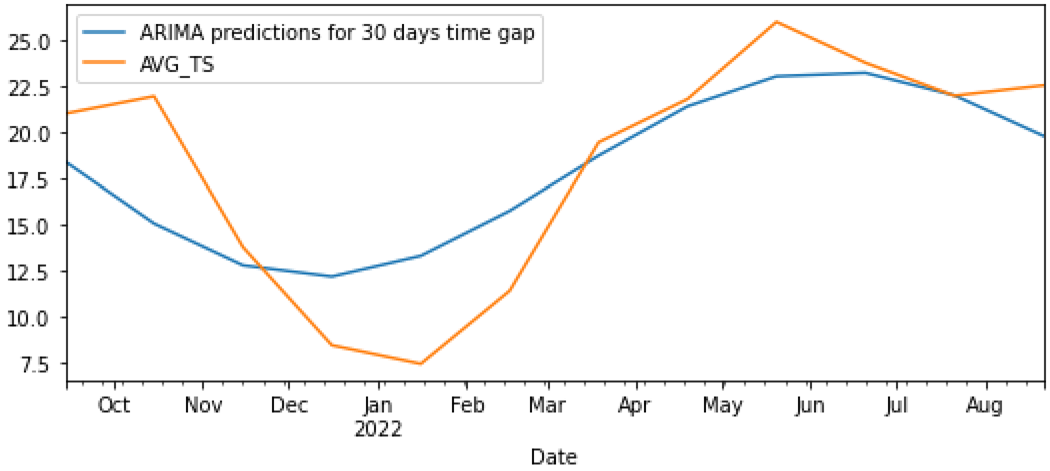


Fig 5.5: ARIMA Prediction with 30 day time gap

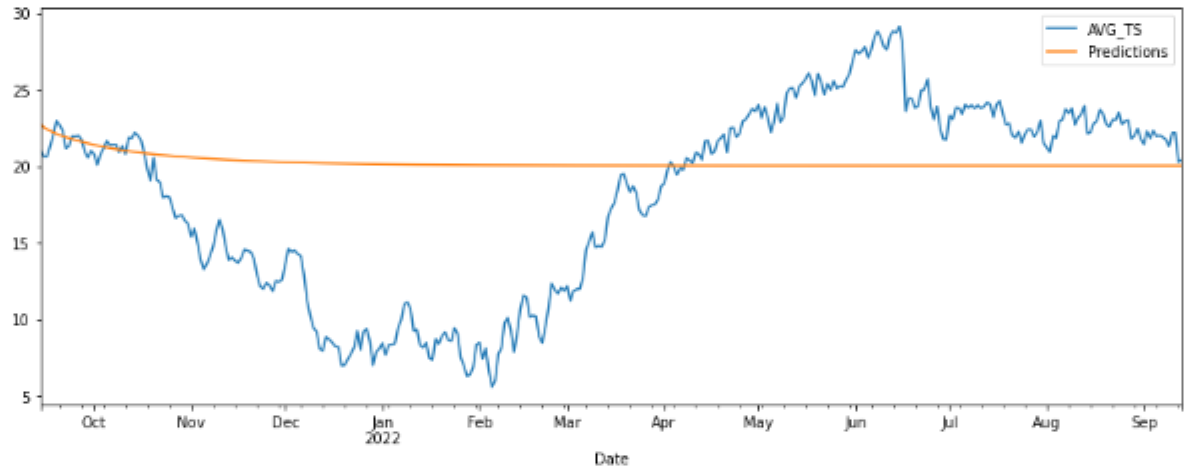


Fig 5.6: LSTM Prediction with no day time gap

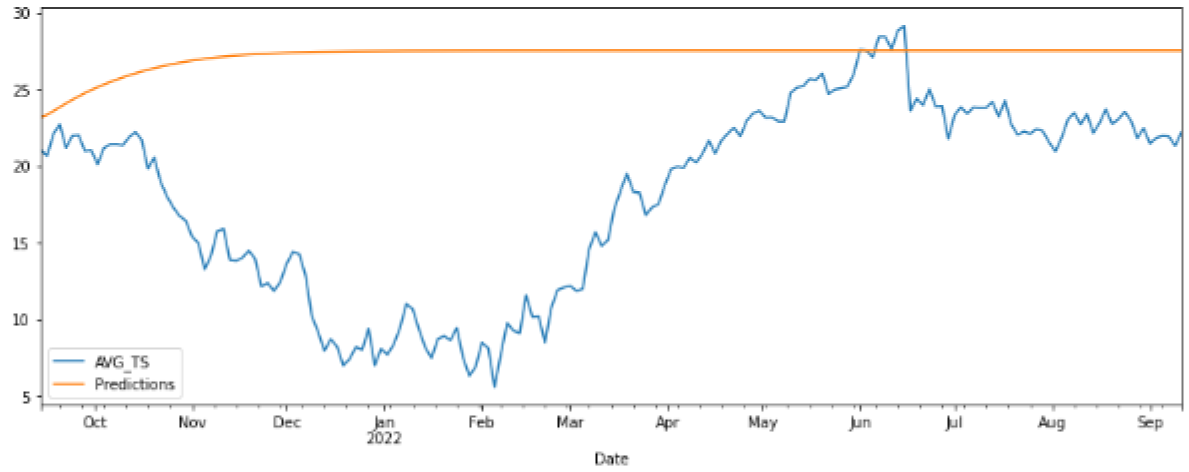


Fig 5.7: LSTM Prediction with 1 day time gap

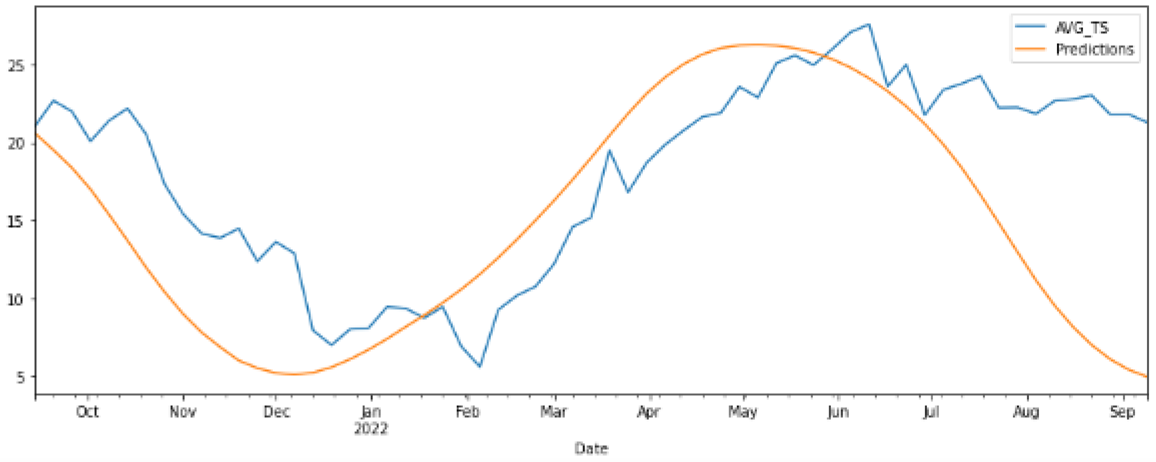


Fig 5.8: LSTM Prediction with 5 day time gap

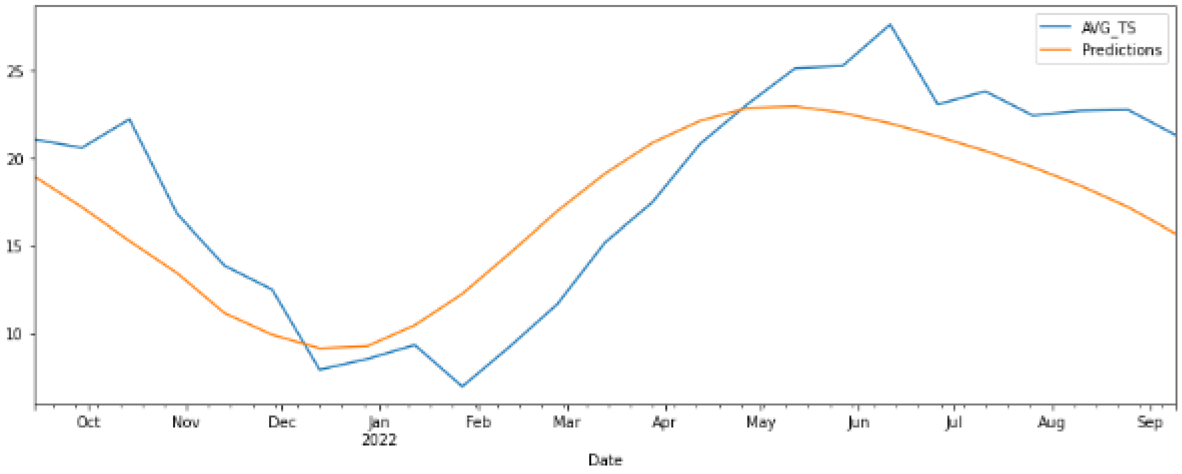


Fig 5.9: LSTM Prediction with 14 day time gap

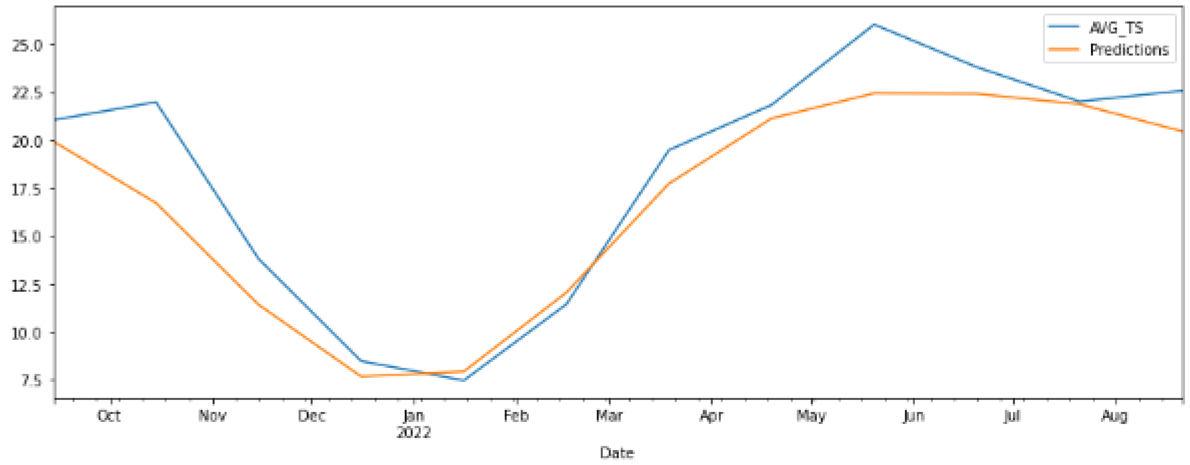


Fig 5.10: LSTM Prediction with 30 day time gap

To compare how good the model is doing, we have taken mse/rmse (root mean square error) and r2 square as our metrics. Even though based on a rule of thumb, it can be said that RMSE values between 0.2 and 0.5 shows that the model can relatively predict the data accurately. In addition, Adjusted R-squared more than 0.75 is a very good value for showing the accuracy. In some cases, Adjusted R-squared of 0.4 or more is acceptable as well. But we can take here lowest RMSE and r2 square more than 0.75 as a good model while comparing.

RMSE ERROR COMPARISON		
Day Gap	LSTM	ARIMA
0 day gap	6.5675	7.248809
1 day gap	11.08280	3.03237
5 day gap	6.70079	2.93082
14 day gap	3.754016	3.0992
30 day gap	2.2088	3.4093

Table 5.1:RMSE error comparison between ARIMA vs LSTM

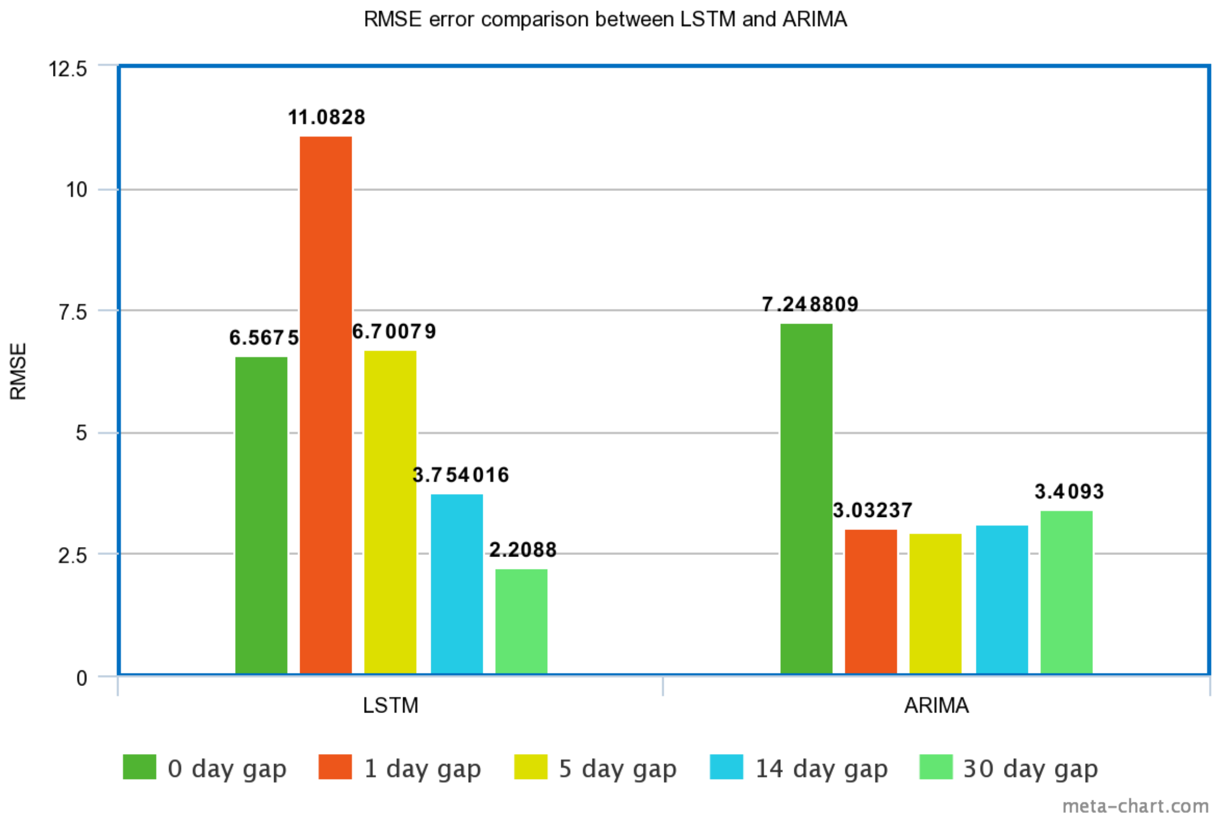


Fig 5.11: RMSE error comparison between LSTM and ARIMA

R2 SQUARE COMPARISON		
Day Gap	LSTM	ARIMA
0 day gap	-0.11150	-0.3540
1 day gap	-2.1727	0.762481
5 day gap	-0.18399	0.77349
14 day gap	0.63072	0.7483
30 day gap	0.86623	0.6813

Table 5.2: R2 Square comparison between ARIMA vs LSTM

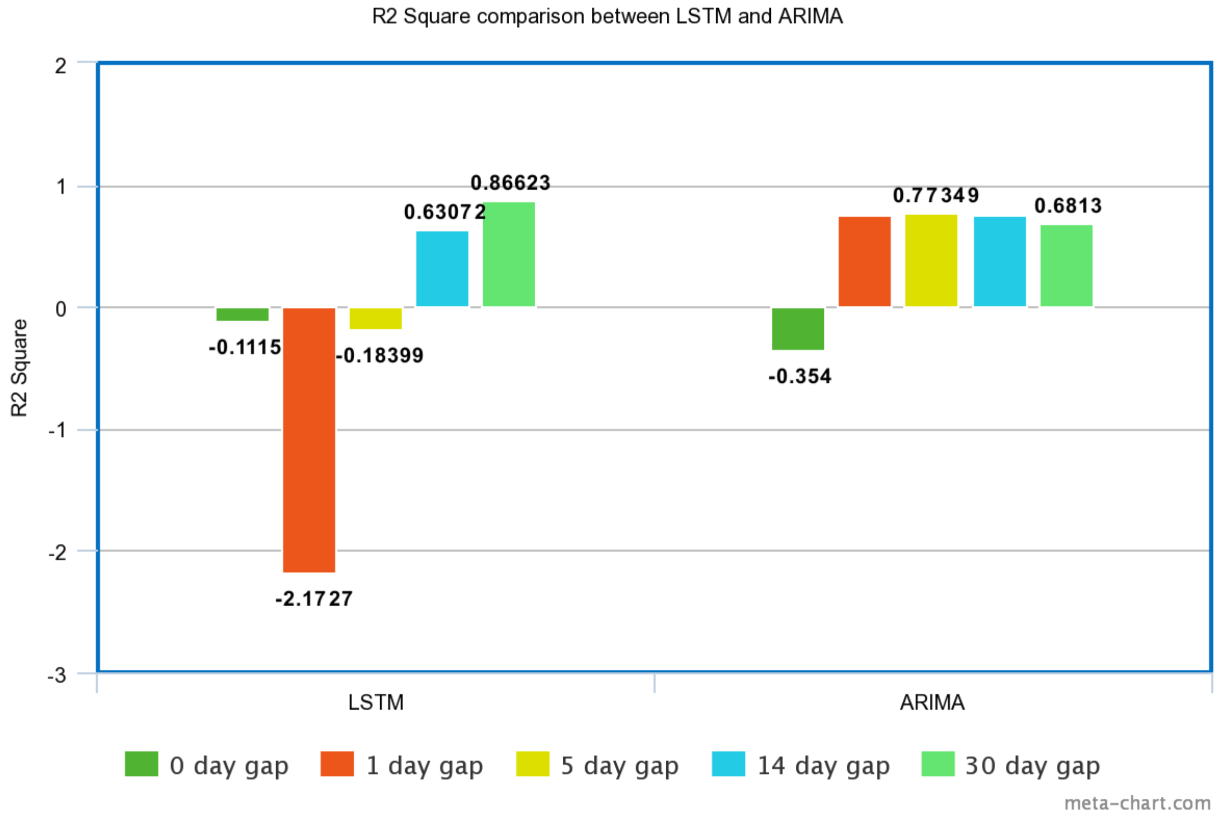


Fig 5.12: 16 days prediction graph with respective time gap ARIMA vs LSTM

From the above given data we concluded that ARIMA model produced lower error values than LSTM model in monthly and weekly series which indicated that ARIMA was more successful than LSTM for monthly and weekly forecasting. While the error values produced by LSTM were lower than those by ARIMA for daily forecasting in a rolling forecasting model. Both ARIMA and LSTM predicted the seasonal fluctuation well, particularly in rolling forecasting models.

The patterns of average surface temperature of Kathmandu valley is suitable for the ARIMA model and LSTM model. However, the different principles of these two models resulted in different performances at different time scales. The principle of the ARIMA model is to filter out the high-frequency noise in the data, detect local trends based on linear dependence and predict the development trends. The limitation of this model is that ARIMA can only analyze the linear part of a temperature series. However, the non-linear part of temperature data may not be white noise, meaning that some information may not be captured by the ARIMA model. LSTM is an advanced kind of RNN and a deep learning application which is designed to learn temporal patterns, capture nonlinear dependencies and store useful memory for a longer time so produces better results in situations where the number of dataset is large. In our study, according to the above fig the daily series had the largest number of data points and the most non-linear dependencies while the monthly series had the smallest number of data points and the most linear dependencies. The results show that the ARIMA model tends to forecast more accurate results for which there is a clear trend in the series, whereas LSTM tends to do better on volatile time series with more unstable components. In addition, the ARIMA model and LSTM model have different requirements for sample size. ARIMA model needs statistical inference in the process of modeling, so it needs to meet the requirements of large samples. Studies have shown that ARIMA needed at least 50 historical statistics. The LSTM model is a complex neural network, and like any neural network requires a large amount of data to be trained on properly. Too few training samples will lead to over fitting. The larger the sample size, the higher the accuracy of the model. Therefore, the LSTM model is not recommended when the sample size is too small, such as monthly or weekly data. The above might be the reasons why ARIMA showed better performance in monthly and weekly predictions while LSTM displayed better performance in daily predictions.

Chapter 6: Conclusion

Both ARIMA and LSTM could be adapted to build prediction models for the average surface temperature .The best fitting model of ARIMA and LSTM were adopted to forecast the temperature ,direct forecasting method was used while prediction .Predicted values were compared with the actual values to test the prediction effect of the models. RMSE was applied to evaluate the prediction performance of the models and lower value means better performance.. ARIMA model produced lower error values than LSTM model in monthly and weekly series which indicated that ARIMA was more successful than LSTM for monthly and weekly forecasting. While the error values produced by LSTM were lower than those by ARIMA for daily forecasting in a direct forecasting model.

Chapter 7: Future Enhancement

For now, our model has been trained for only two time series forecasting algorithms , but in future we can increase the number of algorithms for better comparison . We can also train the algorithms for multiple data to forecast the temperature in a more accurate manner .

References

1. Goswami, Kuldeep,. “Monthly temperature prediction based on ARIMA model.” *International journal of advance research in computer science*, vol. 8, 2017. *researchgate*,

2. Agrawal, Raghav. “Time Series Forecasting.” *Analytics Vidhya*, 2021,

3. Artley, Brendan. “Time Series Forecasting with ARIMA , SARIMA and SARIMAX.” *Towards Data Science*, 2022,

4. Brownlee, Jason. “A Gentle Introduction to Long Short-Term Memory Networks by the Experts.” *Machine Learning Mastery*, 2021,

5. Lai, Yuchuan, and David A. Dzombak. “Use of the Autoregressive Integrated Moving Average (ARIMA) Model to Forecast Near-Term Regional Temperature and Precipitation.” *Weather and Forecasting*, vol. 35, no. 3, 2020. *American Meteorological society*,

6. NASA. *Data Access Viewer*,