**Forecasting the prices of natural gasses using ARIMA - LSTM - BLSTM - Neural Prophet deep learning methodology**

**Devasenan M | Shivaramakrishnan R**

Published online:

**Abstract:**

Natural gas, entitled as methane gas or natural methane gas, is a highly flammable, colorless, odorless gaseous hydrocarbon where ethane and methane forms the core. A petroleum resource which is associated with crude oil, burning it results in less emission of carbon which promotes a sustainable environment. In order to be extremely safe, go ecogreen, reduce dependency on nations for fuel resources it is even more tactical to forecast the prices of natural gasses in the international market for a time frame. Thus, an ARIMA model is developed initially for the forecast. The autoregression predicts the upcoming values based on then values. Moving Averages play a crucial role in smoothing the time series data. Secondly, the LSTM model is constructed with the same dataframe. LSTM uses recurrent neural networks (RNN). The ideology behind the model is that at times we have to be conscious of recent information to perform the present task. Bidirectional LSTM is also constructed. The Neural Prophet which is built on the top of pytorch is also experimented by means of forecasting. Neural prophet is extensively used by developers for the extension of the framework. The experimental repercussion showed that the proposed models are more efficient in terms of prediction and accuracy

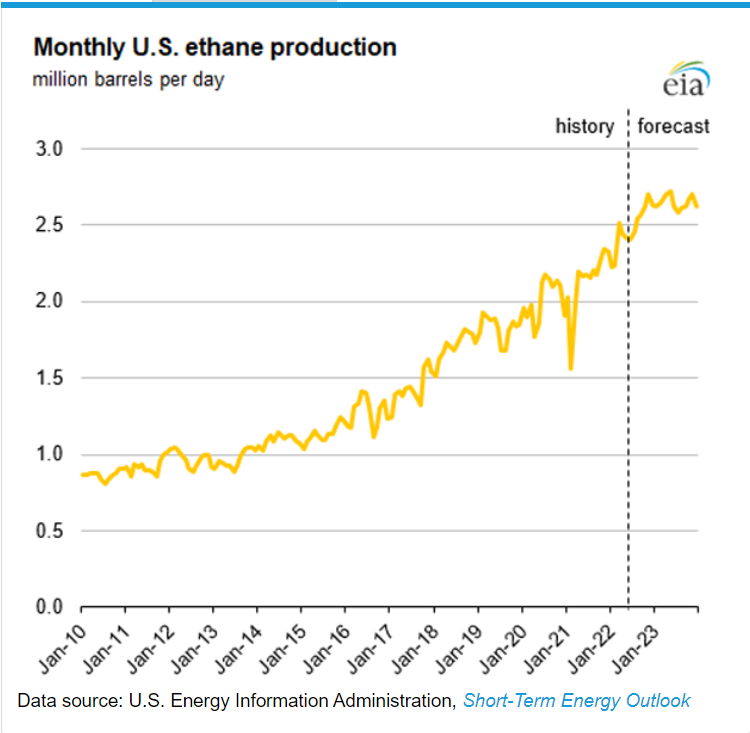
**1 Introduction**

**1.1 Background**

Hydrocarbons which are formed beneath the earth’s surface are natural gasses. The level of natural gasses is adequate for around 230 years, if the consumption remains at current levels (IEA). As gas is the only cleanest burning hydrocarbon, it emits carbon-di-oxide partly. Natural gasses are primarily responsible for a quarter energy consumption in the States (Trading Economics).The prices of the gasses eminently flucatues in the period of winter and the States is yet another reason for a great disparity. The need for natural gasses is going through the roof at most recent times.

When it comes to power generation, there is a wide ultimatum for natural gasses as the production increases persistently. Hence natural gasses can be a booming factor in the international market in the nearing days. Ethane production at U.S. gas processing

plants, which has been growing annually since 2017, rose to just above 2.5



**Figure 1 shows the monthly U.S. ethane production**

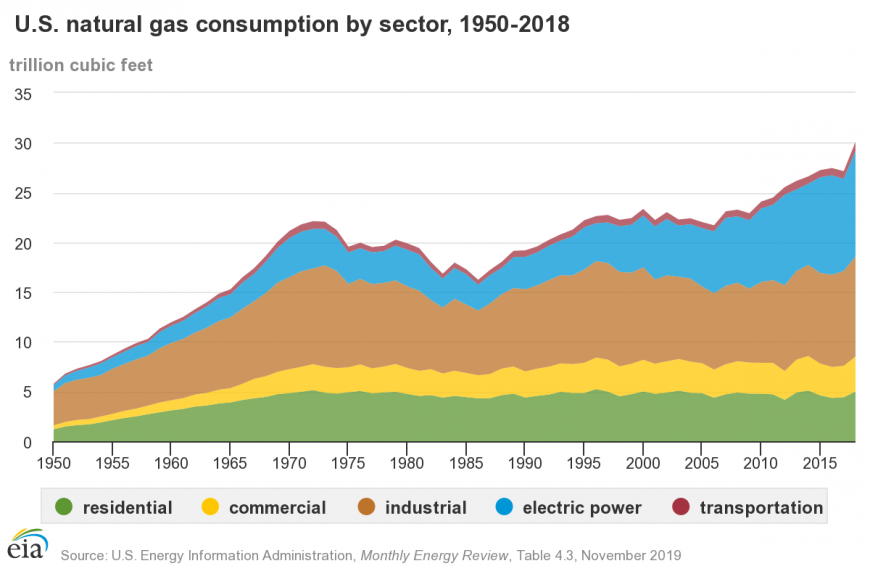
million barrels per day (b/d) in March, and has remained above 2.4 million b/d since then. As the States remains the leading producer of the natural gasses, our forecasting models will be a leading tactic in the aspects of investment and formulation. A serious conflict began in 2005 between Ukrainian oil and gas company and Russian supplier over prices, logistics and supply (Economic Times). A small business dispute later turned into a political war. In addition to that, the volume of the gas is about 600 times smaller than its volume in its gaseous state (U.S EIA). Also natural gasses have a variety of uses. It can be used in the field of power sectors for heat generation, ammonia in the fertilizer sector and CNG in the transport sectors. Hence the accurate forecasting of the prices addresses various disputes, helps in the advancement of investment and planning.

**1.2 Impulse**

Natural gas is an abundant fossil fuel. Unlike other fuels, natural gasses are formed from plants, animals and microorganisms that lived millions of years ago. Deposits of the gasses need not be much deeper. The deposits are often found near oil deposits. Such deposits are dwarfed easily by the nearby oil refineries (National Geographic). Electricity can be generated using natural gasses with the help of steam turbines and gas turbines.

It also can be converted from CNG to LPG and is used for industrial purposes.

Many developing and developed countries are dependent on petroleum exporting countries (OPEC) for the fuel resource. The cost of importing (transportation) petroleum resources is higher than the resources. Also, the storage units of the gasses are limited to a capacity and hence our forecasting will help to store according to adequacy. The holy grail behind the forecast is the conspiracy of natural gasses in this supremacy world.



**1.3 Existing Challenges**

Natural gas price forecasting technology has been for over three fourth of a decade. Some of the key challenges worth to be addressed are:

(1) Picking the right features from the available dataset and feature engineering.

(2) Improving the forecasting accuracy of the existing models.

(3) Handling the non-stationarity of the given data in particular to the designed architecture.

(4) Handling the regular patterns observed (seasonality) in the price variation throughout the year.

(5) Using multiple dependent variables for accurate prediction

(6) The price of maneuvered dataset is entirely a nation based and not a wholesome

**1.4 Endowment of the constructed model**

Looking into the above concerns, a number of models have been constructed which are further more coherent and meticulous, the models are trained, tested and are showcased in this paper.

**Contriving models**

1. Autoregressive Integrated Moving Average model (ARIMA)
2. Long - Short Term Memory model (LSTM)
3. Bidirectional LSTM model
4. Neural Prophet Model

* The presented paper avails data preprocessing and data scaling using Min Max scaler for better optimisation. In this model, the total number of lags or differencing amount applicable to the data and the stationarity is verified. The resultant outputs are interpreted to that of multiple regression models.
* Long - Short Term Memory is a noteworthy model RNN which is eligible for swotting long term data dependencies. This achievement of the model is due to the interaction of the combined layers. Persistent memory is one of the use cases of this model. It is a pro handler of vanishing gradient problem handled by RNN
* In bidirectional LSTM, in spite of training a single model, we introduce a second model. The first model learns the provided input sequence whereas the second model learns the reverse of that destined sequence. It also increases the information amount improving the algorithm. However the bidirectional neurons do not interact with each other.
* Neural Prophet increases the model accuracy, where components of some model can be marshaled into neural networks. This model uses hyperparameter selection which adds more compatibility to the developers. Prophet always marks standards of industry. NeuralProphet reforms Prophet from the bottom up, stressing the challenges with replacement of Stan with [PyTorch](https://l.facebook.com/l.php?u=https%3A%2F%2Fpytorch.org%2F&h=AT2qzuqrKUY1Ct2NXy08jcM31G3cjTjCf0t5uBJ8lUgKo3yqUPywJZeUcyF7GAKQGgHRvPYkhHOVm4wam2S5lYTw9Bmm5lJDCU0SdE55-SYkVTJUd_sKhNAAOXM0Kt6jJY4g0nTTXgGdCCiXzJxyYQ), which is both easy to use and pliable. This helps in the advancement of the developers to add upon new features with their extended framework and to adopt into new research technologies.

**2 Related works**

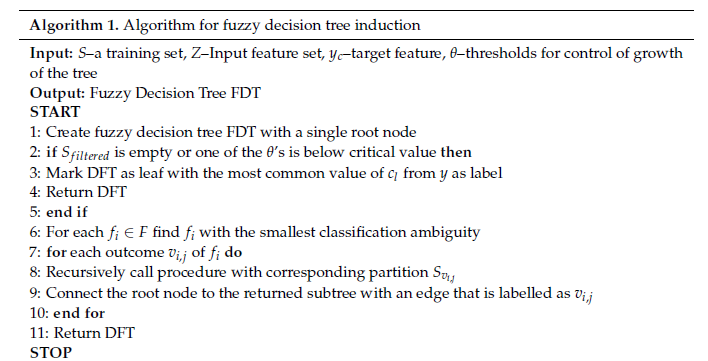
**2.1 Natural gas price forecasting**

N number of forecasting models have been proposed and constructed in various research works. The notable algorithms were as follows

* Pruning and Inducing Fuzzy Decision Tree

**Pruning and Inducing Fuzzy Decision Tree**

A tree which involves the selective removal of certain branches of a tree to improve the structure of a tree in order to overcome the problem of overfitting

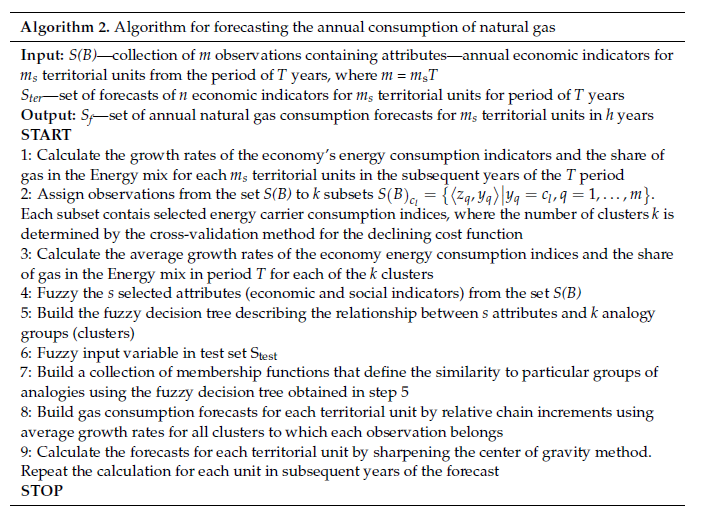


**Fuzzy Decision Tree Learning**

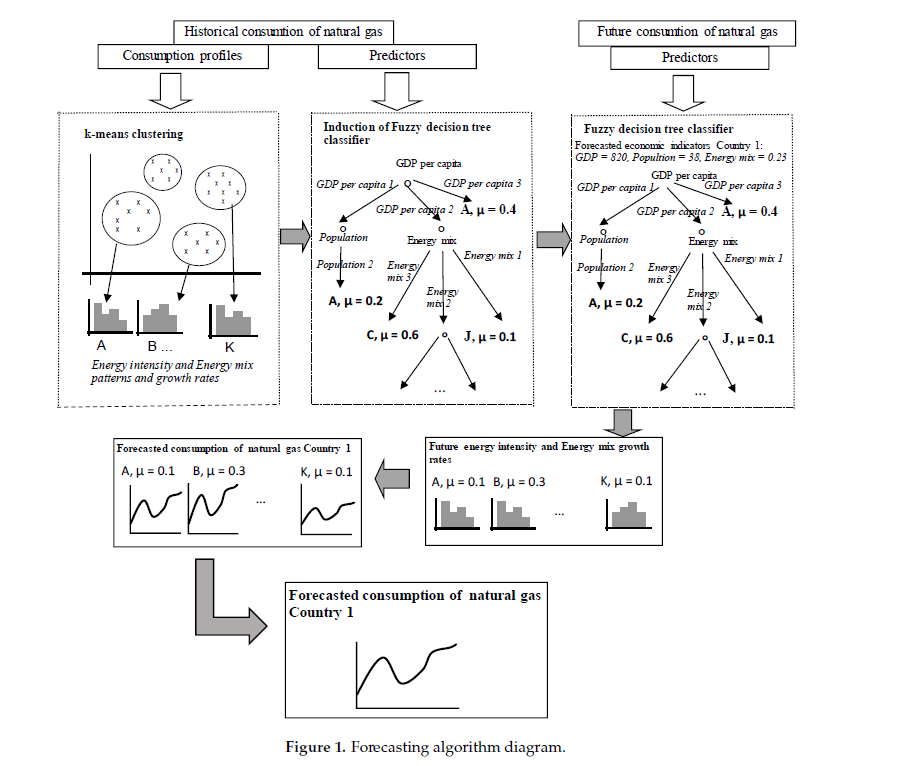
A decision tree which uses a supervised learning method where the assignment to previously created classes is based on explanatory variables. A fuzzy decision tree was built in the next step.

**Steps of fuzzy classification procedure**

1. Converting FDT into set of rules
2. Calculating the membership of the object for the premise of rule for each rule
3. Aggregate membership derived from all rules for each class



The overview of the forecasting is shown below



Thus, the prices of the natural gasses have been forecasted with the above algorithms

**3 The models experimented with:**

**3.1 AutoRegressive Integrated Moving Average (ARIMA):**

ARIMA model is a popular and widely used statistical model for time series analysis and forecasting. In simpler words it is a kind of model that uses time series data either to better draw conclusions from the dataset or to predict the future trends. A time series data is a set of values collected in regular intervals of time, in intervals of seconds, minutes, hours or daily, weekly or even in yearly periods. For instance, an ARIMA can predict the future price of stocks with the help of the past trends and analysis.

Autoregressive models, as the name suggests, for a given dataset it assumes that the future will resemble its past data. A statistical model is said to be autoregressive when it is able to predict the future values based on the given past data.

The name ARIMA stands for AutoRegressive Integrated Moving Average. It is a level up or a better version of the previous simpler model which is quite normal as compared to ARIMA which is just the AutoRegressive Moving Average, where the concept of Integration is included in the ARIMA model.

**Components:**

The ARIMA model is simply a class of standard models that works better together to capture the different temporal structures in the time series data and provides a still powerful method to make future predictions. ARIMA can be better understood by first understanding its core components that form the base to the name Autoregressive Integrated Moving Average.

* AutoRegressive (AR): The past data in a time series data can create an impact on both the current and future values. So ARIMA inherits this concept of considering past values while forecasting the future values. As previously stated, autoregressive or autoregression refers to a model that uses the dependent relationship between an observation and its own previous or lagged values.
* Integrated (I): Integrated refers to the differencing to be done on the raw data to make the time series data stationary. Differencing is an operation done on the time series which is the difference of an observation and its previous observations.
* MovingAverage (MA): It is a model that uses the dependency of an observation with its residual error computed from a moving average model applied to lagged observations.

**Parameters:**

To build an ARIMA model, we need to consider the functional parameters that are used in defining the above-mentioned components of an ARIMA model. ARIMA uses a standard notation of representing these parameters namely p, d, and q. Each of the parameters represents its respective component in the model.

* p: Is known as the lag order, which respects the AutoRegressive part of ARIMA. p is the number of lag observations that is considered for the model.
* d: Is known as the degree of differencing, which respects the Integrated part of ARIMA. It is the number of differencing we perform on the given raw data. To make the raw observations stationary.
* q: It is known as the order of the moving average, which respects the MovingAverage part of ARIMA. It is the size of an average moving window.

**Flexibility of parameters:**

An ARIMA model is defined using these primary parameters and (p,q,d) is known as the order of ARIMA. But, doesn’t mean that the ARIMA order cannot have any zero values. (p,d,q) can have any zero values, which implies that the respective component of ARIMA should not be used. The ARIMA model can be built simply to perform just the AR, I, MA separately or ARMA models. For instance, If the given time series data is observed to have no trend or any seasonality, it is considered to be stationary. So it is not necessary to make use of the Integrated component (I) of ARIMA.

**ARIMA and Stationarity:**

To make sure we use the full potential of an ARIMA model, it is important to check for stationarity in the time series data. A time series which shows stationarity is the one that implies that the data is consistent over time. A non-stationary time series data performs worse than a stationary time series and it could make a huge impact on the accuracy of the results. So it is one of a important steps in building an ARIMA model to check for stationarity in the given time series data. If not the data is different, that is subtracting an observation from another observation at the previous time interval, in order to make the time series data stationary. So the purpose of differencing a time series is to remove any kind of repeated trend or seasonalities observed in the data.

In this step we would come up with a value for the parameter d. This is the first and foremost step involved in building an ARIMA model.

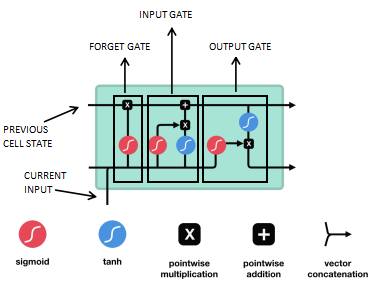
**Identify P and Q:**

Once the differencing is done for a Non-stationary data, the next step is to identify the order of p and q parameters for the AutoRegressive and MovingAverage terms. There are a set of primary tools for identifying the values of p and q.

**3.2 Long - Short Term Memory model**

LSTM is a recurrent neural network model predominantly used in case of sequence prediction problems that has feedback connections. Also capable of processing entire data sequences rather than single data points such as texts, images,etc.. The parts of LSTM are known as gates. The gates are

* Input Gate
* Output Gate
* Forget Gate



The first and the foremost part chooses whether to remember the information from the precursory timestamp is relevant or to be remembered. In the event of irrelevance, it can be forgotten. The second segment, the cell will be trying to learn new information from the input. Finally, in the last segment, the cell transfers the information updated to the next timestamp right from the current timestamp.

**Sigmoid:**

Sigmoid activation is similar to that of tanh activation. Instead of pushing the values between -1 and +1, it pushes the values between 0 and 1. Helpful in updating the data making it to forget as anything multiplied by 0 produces a 0 always. Anything which is multiplied by 1 is always a 1 making the data to be remembered. Therefore, the network can easily learn to distinguish among which data to remember or forget.

**Activation of tanh:**

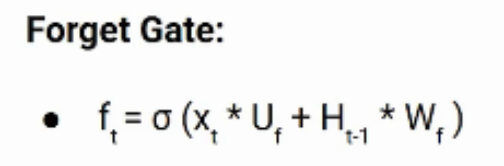
The vectors flowing via a neural network undergo many transformations due to various operations of mathematics. Tanh activation is extensively used in regulating the values flowing through the network. The internal operations of tanh are very less but works fine with the given circumstances. It pushes the values between -1 to +1

**Forget Gate:**

The gate which decides whether information should be kept or thrown away. Information from previous hidden state and from current input is passed via sigmoid function. The value ranges between 0 and 1.

0 or nearly closer - to forget

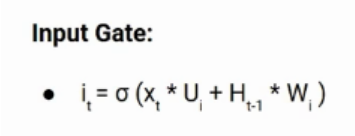
1 or nearly closer - to keep



* Xt - Current timestamp input
* Uf - Weight matrix of input weight
* Ht-1 - Hidden state of previous timestamp
* Wf - Weight matrix of hidden state

**Input Gate:**

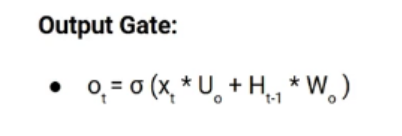
Extensively used in quantifying and qualifying the significance of new information which is carried by the input. The equation of the input gate is given below.



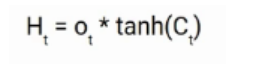
* Xt - Input which remains at the current timestamp t
* Ui - Weight matrix of the remaining input
* Ht-1 - Hidden state at the preceding timestamp
* Wi - Weight matrix

**Output Gate:**

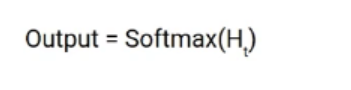
Output gate ranges between 0 and 1 due to the sigmoid function which decides what the next hidden be of. Previous hidden state and the current input state is passed into a sigmoid function. Nextly, the newly modified cell state is passed to the tanh function. Multiplying the tanh output and the sigmoid output gives a clarity on what information the hidden state should carry.



Current hidden state is calculated using the formula given below

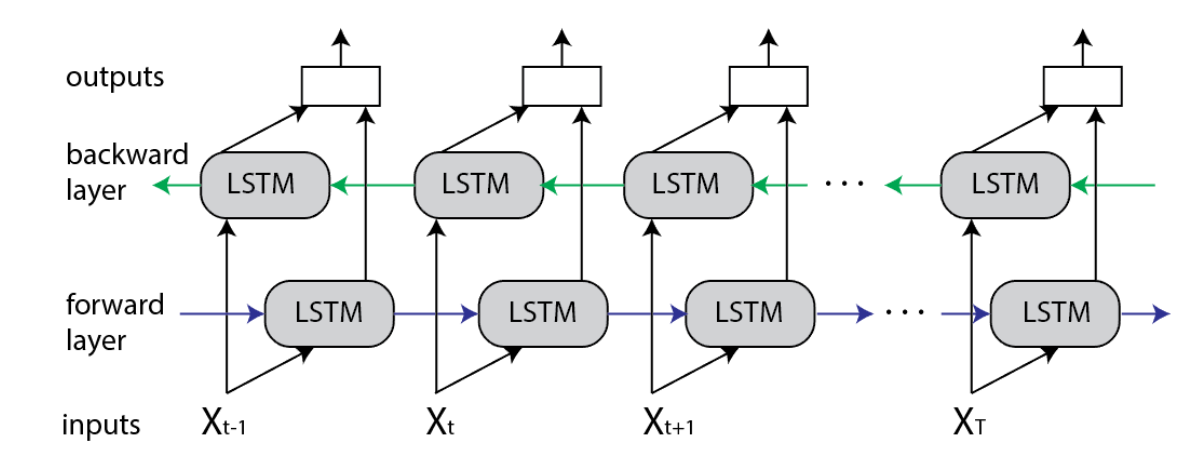


Output of the current timestamp is taken by SoftMax activation on hidden state



**3.3 Bidirectional LSTM**

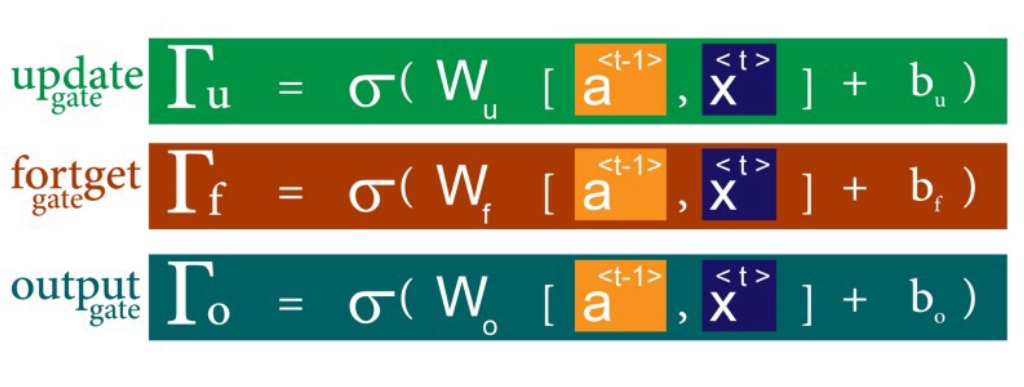
The ideology behind bidirectional LSTM is straightforward sequencing. Moreover, it is the formation of neural networks in both forward and reverse direction. In the streamlined LSTM, the input can flow only in one direction either forward or backward.



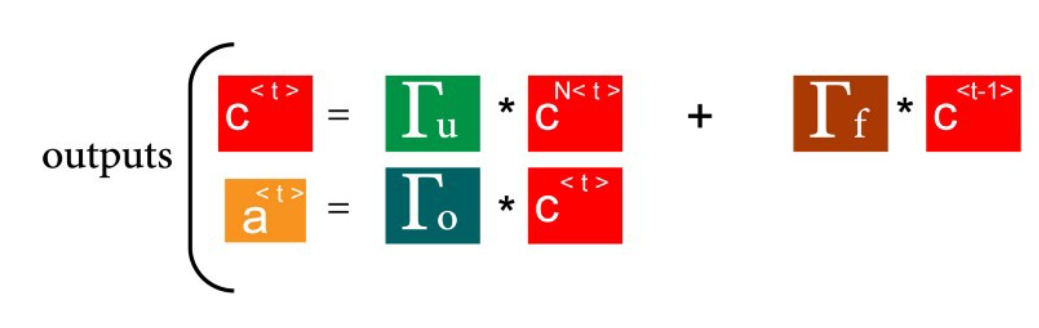
The above diagram describes the flow of information from forward and backward layers. The bidirectional LSTM are activated when the sequence to sequence tasks are required.Two activation values are used in this type of LSTM and also we have two outputs from the cell, a new activation, and a new candidate value. The formula for calculating the candidate value is shown below.



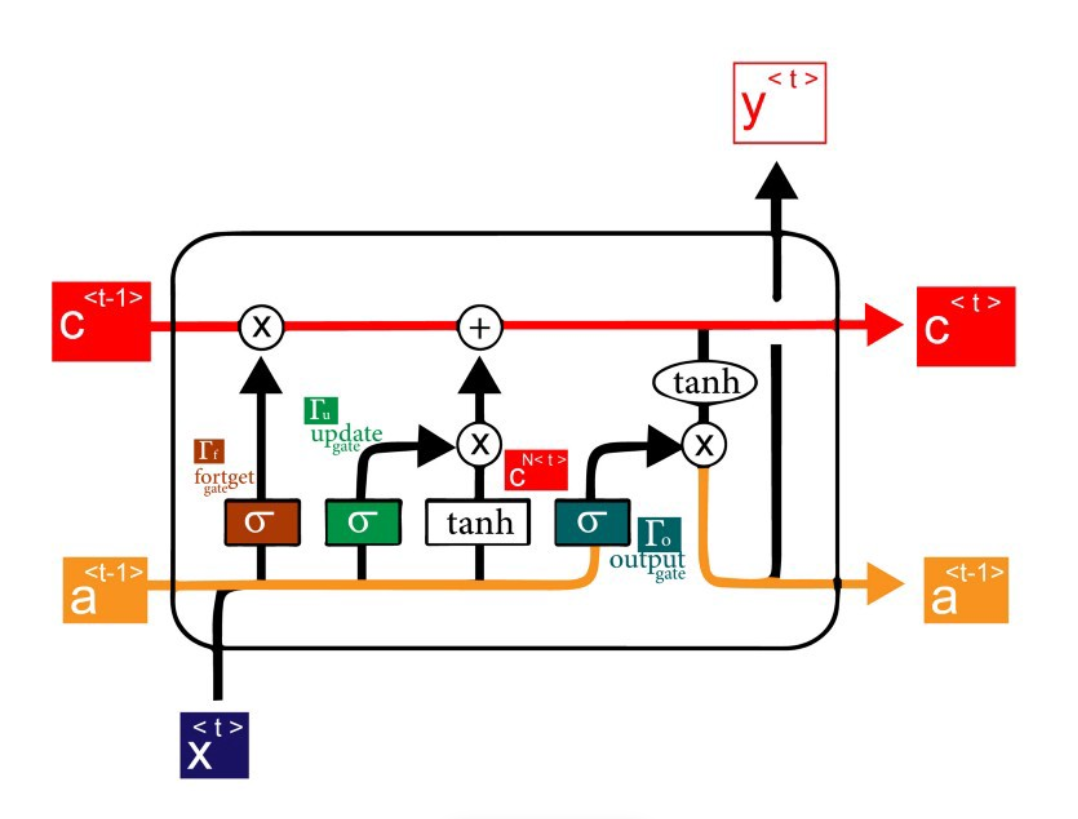
The 3 gates which acts as the mastermind are



Since we have two outputs from the LSTM, outputs are given as

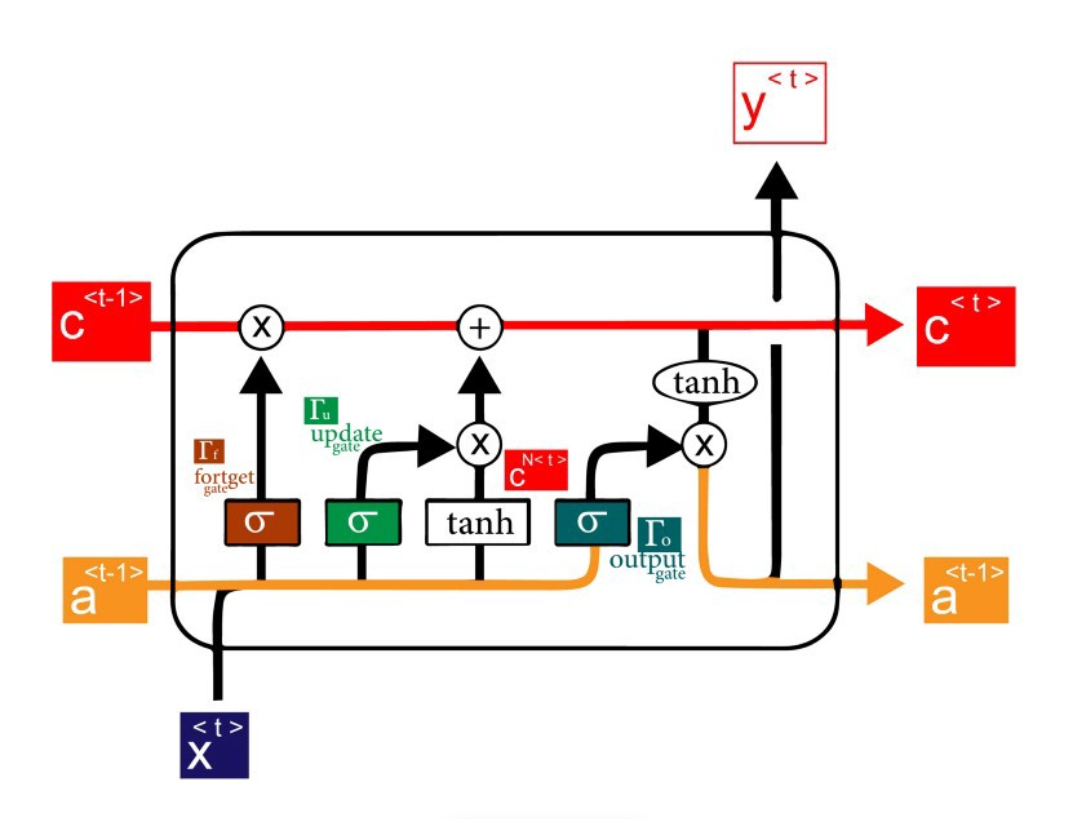


**Combined Output:**

****

**Advantages:**

* Each component of the input sequence has both present and past information.
* Capable of producing more meaningful output from both directions
* Used in speech recognition, text classification

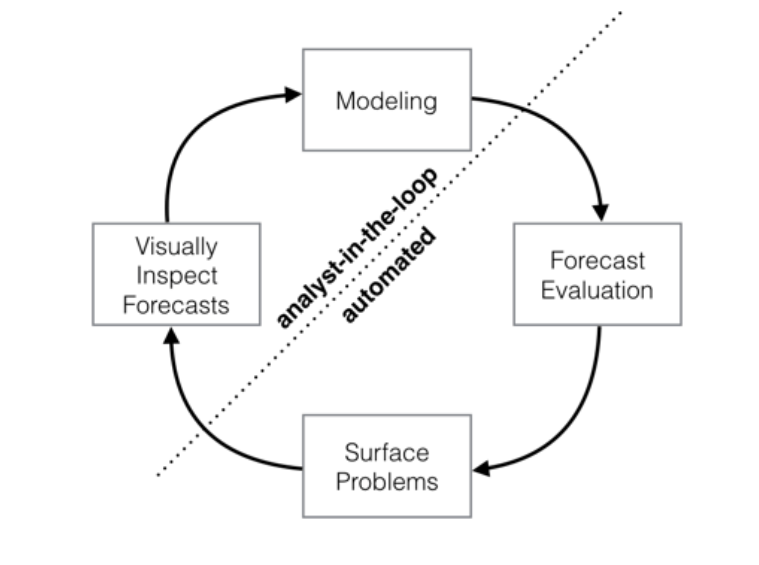
****

**3.4 Neural Prophet**

An open source library module exclusively used for time series modeling and was introduced by facebook. Based on pytorch, this is an improvised version of the FB Prophet model. It highlights usual cases including scalability and extensionality.

Can also be used in anomaly detection.

The introduction of neural prophet in addition to prophet envisioned frontline engineers and industrial leaders to improvise a wide range of industrial applications. Future values must depend on past observations.



**Prophet VS Neural Prophet:**

* Local context for predictions are missing in prophet whereas Neural Prophet supports auto-regression and covariates
* Difficulty in extending framework whereas pytorch will be helpful in extending framework in neural prophet

**User Friendly Package**

1. A leaning curve which is smooth and gentle
2. Improvise and improve according to the received results
3. Extremely powerful
4. Easily customisable
5. Widely expandable

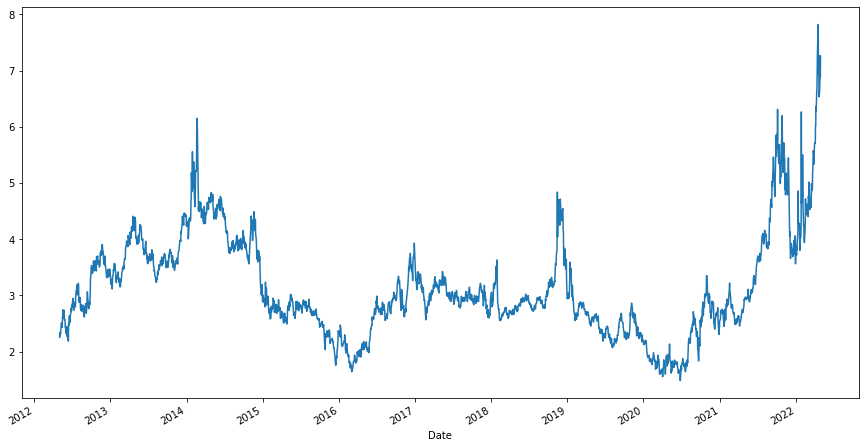
**Hyperparameters**

* The default loss function is Huber loss unless it is user defined
* The epochs and batch size are approximated with the size of the dataset
* As an optimizer for simplicity, we use one-cycle policy with AdamW

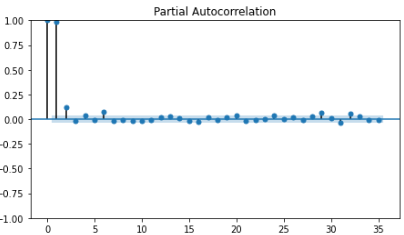
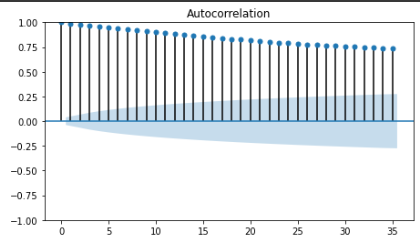
**4 Construction and Implementation:**

**4.1 ARIMA:**

In this section, we can plot a line graph first to analyze the time series.Here, we just take the closing price as target and with which we train the model. This ARIMA is a Univariate model.



As, we can interpret that the above graph shows a trend, which implies the data follows a seasonality. Where the data is non-stationary. To be more precise, the seasonality in a time series can be identified using a autocorrelation plot and partial autocorrelation plot. ARIMA models work only when the given time series data is Stationary.



Here we can see that both the ACF plot and PACF plot, has got some periodical pattern, which is very likely to follow some seasonal component.

Also there is another method, for concluding that the data is stationary or not. There is a method to test for stationarity of a given time series using an adfuller test.

Augmented Dickey-Fuller is a test done to check whether a time series data is stationary or not. It is a Hypothesis test stating a Null Hypothesis that a root exists for a AutoRegressive time series data. If the result of adfuller test is less than or equal to 0.05, we can reject the Null hypothesis and conclude that the time series data is Stationary.

The test is done for the time series that we have chosen and the result happens to be concluding that the data is Non-Stationary. So, it is confirmed that the data follows some seasonality and it is obvious to do the differencing on the time series.

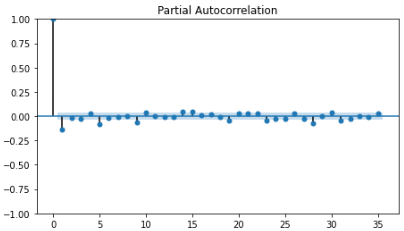
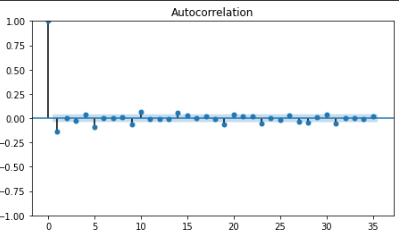
Picking the right differencing order or the d value totally depends on the nature of seasonality the time series follows. Once we get the d value, we can proceed with finding the other 2 parameters. There are two ways to find the differencing value, either we can observe the nature of seasonality in the time series using plots, or we can also use the pmd arima package to get the number of differencing. For our time series, we got the number of differencing to be done as 1. So the d value can be set as 1.

The next step is to plot ACF and PACF plots in order to find the other two parameters p and q.

**ACF and PACF plots:**

* ACF: ACF stands for the Autocorrelation Coefficient Function, which defines how the time series data are correlated to its previous values. It is used to determine the q term in the ARIMA order.

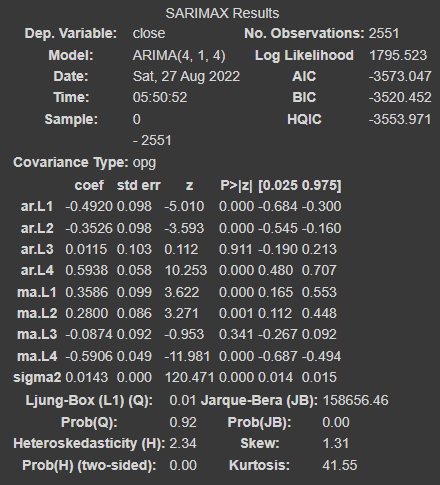
* PACF: PACF stands for the Partial Autocorrelation Coefficient Function, which determines the p term of ARIMA order.



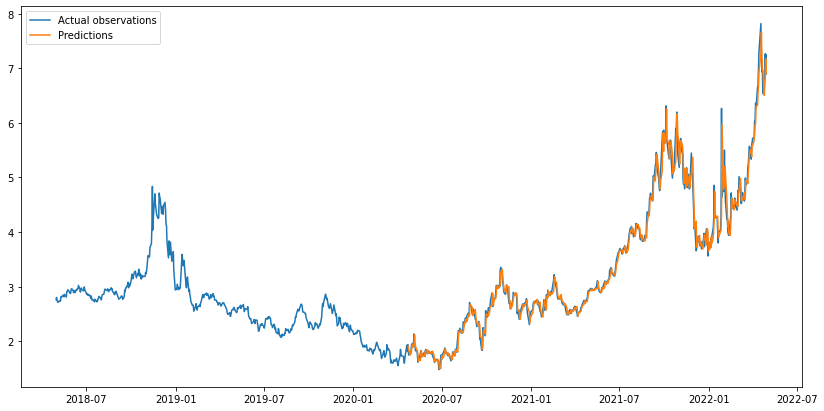
This is the ACF and PACF plots after the differencing is performed on the time series. Now, we can see that both the ACF plot and PACF plot decay very fast, so we can infer that the data is made stationary. We can accept the plot if only less than 5% of dots lie outside the blue shaded region.

When looking at both the ACF and PACF plots, we should ignore the lag 0 which has the long spike. The cut off at ACF plot will give the order q for moving average part. And the cut off at PACF plot will give the order p for the autoregressive part.

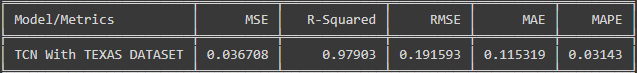
Since the both ACF and PACF plots have been cut off after lag 4, we can set value 4 for both p and q metrics. Now we can use the ARIMA order(p,d,q) as (4, 1, 4).



The predictions have been made for a time series for the past 510 (past 20%) observations..

****

And the results are quite good. The mean absolute percentage error falls to be less than 0.03143, which is quite good for the very shot period of time it takes to train the model.



**4.2 LSTM:**

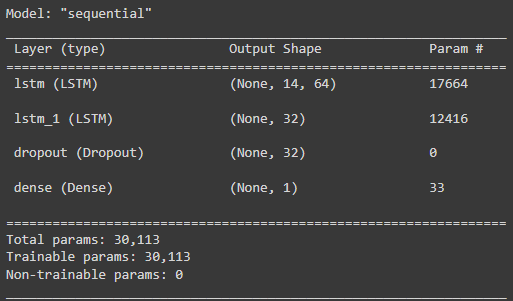
This section analyzes Multivariate time series forecasting done in a LSTM model. For this multivariate forecasting, we take 4 features into consideration for training the model. The closing, open, highest and the day’s lowest price value of natural gas. Though all the data of each feature have values in different ranges, it is important to scale them into the same range.

Before actually defining the LSTM architecture. The data is not prepared for the LSTM model to use. After all the data cleaning, preprocessing and scaling is done. First step is to reframe the time series data for the LSTM model. The structure is just the test length which is the number of previous observations that will be used to predict the next observation. And the future\_length of observations to be predicted by the model.

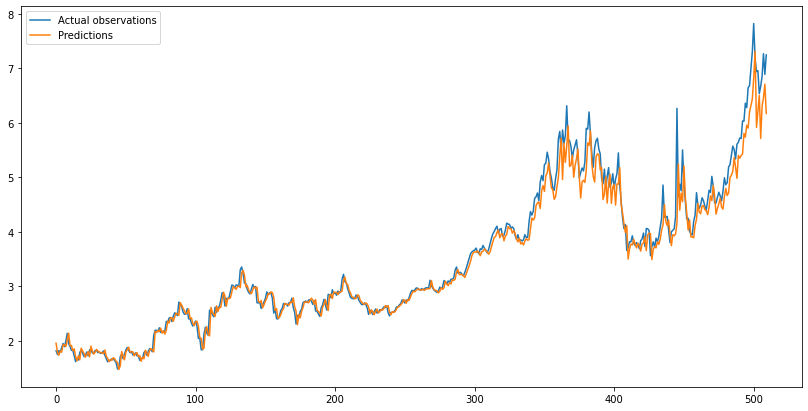
In this case, we use 14 numbers of past observations and try to predict one future observation. With these values the data is transformed for ready to use by the model.

The model proposed here is constructed with an input layer with input shape having 14 observations of all 4 features ( , 14, 4), and two hidden layers of LSTM model both using relu as activation function. One Layer has 64 units of neurons and the other layer has 32 units of neurons. Applied with a dropout layer with a dropout rate of 20% to make sure that the model doesn’t end up with overfitting. The model is compiled using Adam optimizer and MeanAbsoluteError loss.

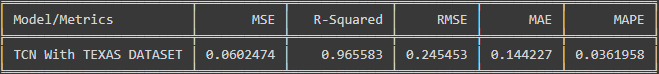
The model is trained with the already transformed training data and the number of epochs is set to 100 with batch size of 16. The batch size plays an important role in the part where the internal state of LSTM in keras does reset itself after each batch. So, the size of the batch is quite important to test with.



The predictions are made same as done for the ARIMA model, for the past 510 (past 20%) observations. After the predictions are made, it is very important to remember that the result of the model, will be a scaled value. So we have to reverse scale it back to normal values.



The results are quite good as well. The mean absolute percentage error is less than 0.0362.

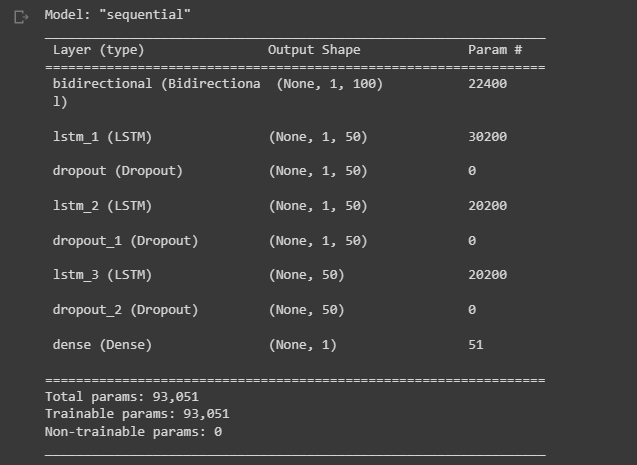


**Bidirectional LSTM:**

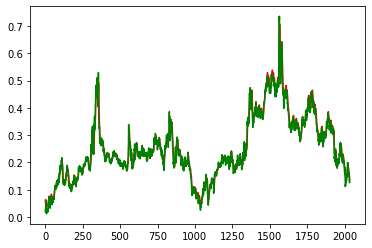
An extension of ancient LSTM which helps in improving the model performance of sequence classification problems. Two directional layers are used in this model. In the first layer, the data is given in the forward layer and in the second layer, the data is given in the reverse direction. As mentioned in the traditional LSTM, before initializing the construction of BLSTM, the data must be processed and scaled using Minmax scaler.

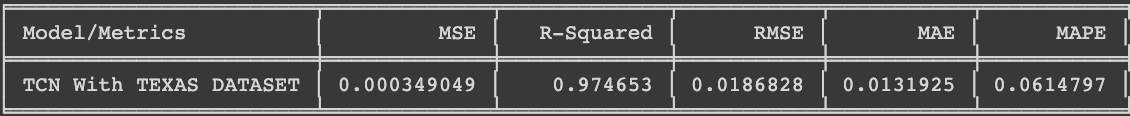
To forecast using BLSTM, four features of the dataset are taken into consideration which are the closing, open, highest and the day’s last price value of natural gas. Though all the data of each feature have values in different ranges, it is important to scale them into the same range. A set of past observations are used for future predictions. In this scenario, we use 5 past observations to predict 1 future observation

The model is constructed by having 5 past observations of all 4 past features with three hidden layers where all layers have 50 neurons each. Dropout layer is added to make sure that the model doesn’t get overfitted. The model is compiled using Adam optimizer and MeanSquaredError loss. The Bayesian prediction is used due to its ability to incorporate prior info to the analysis, and estimates missing values together with the parameter values.The epochs value is set to 50 with the batch size of 5. The size of epoch is to be taken into account as the LSTM in keras resets after each iteration.The summary of the model is given below.



The prediction of the constructed model is shown below

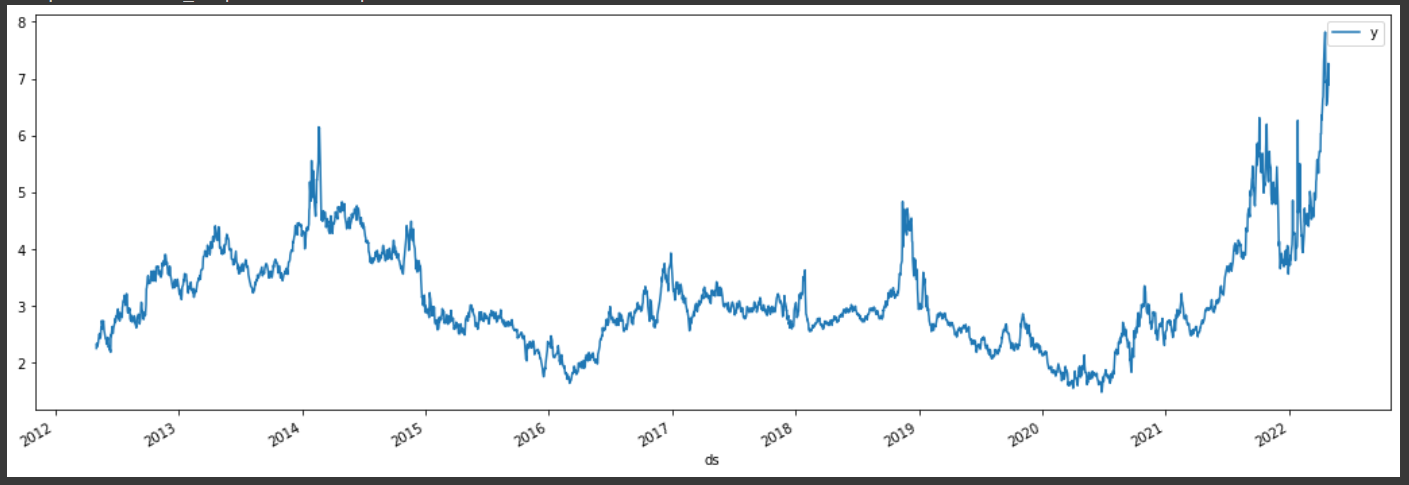




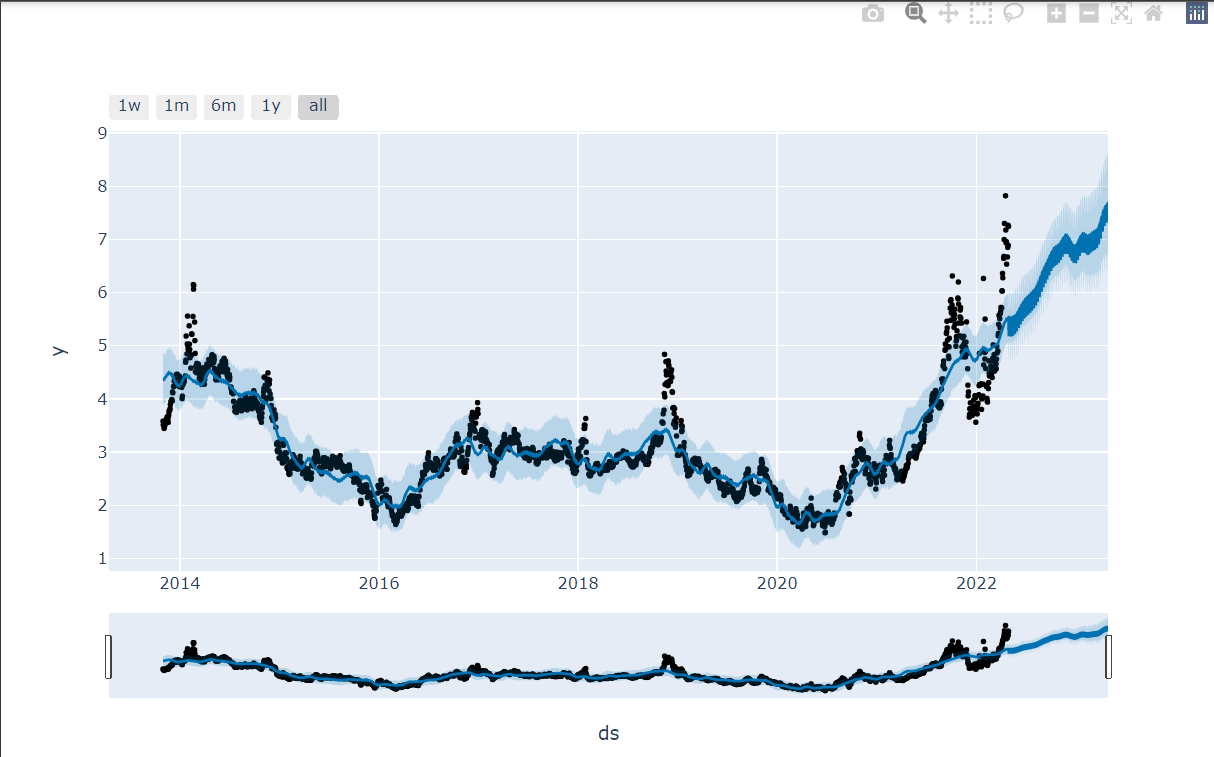
The Mean absolute percentage error happens to be less than 0.0614.

**Neural Prophet Model**

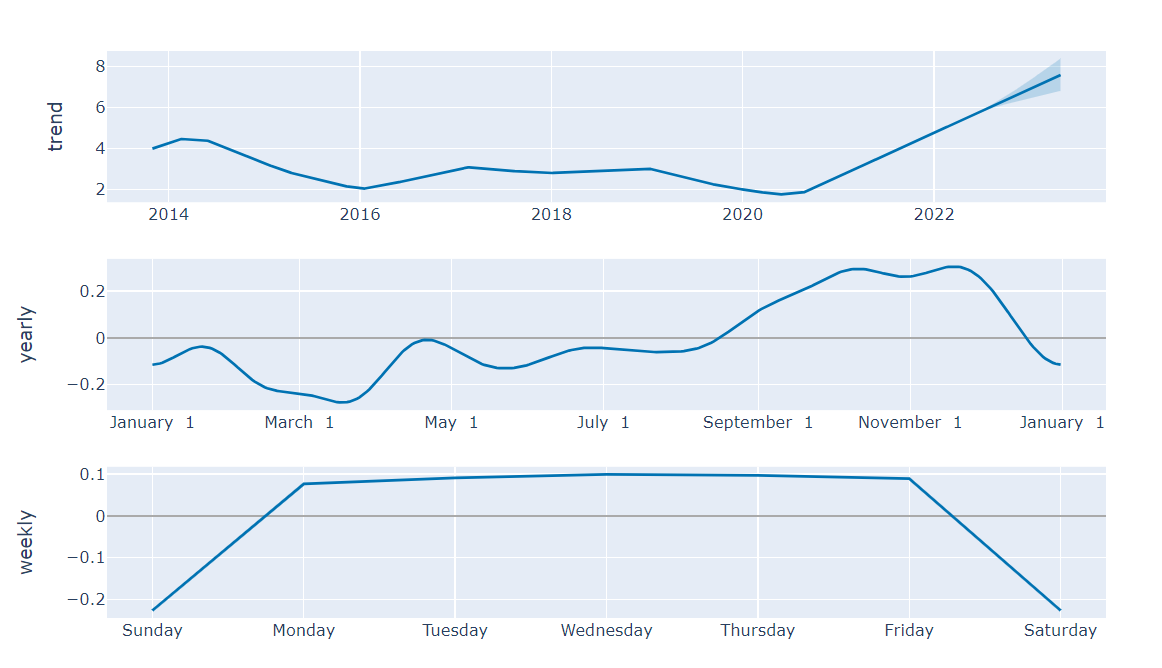
An extension of Prophet model which is inspired by Facebook Prophet and AR-net libraries. Uses gradient descent optimisation for making the model engine much faster. Also uses AR-net libraries for autocorrelation and minimum custom losses and metrics. In addition to above use cases, neural propjet supports covariates which are lagged. The required modules and libraries are installed in the premature state. The data frame is indexed by taking the date and the closing price as the attributes. Now, the indexed dataframe is converted to the proper date-time format. The figure of the dataframe is plotted initially.



Before the actual prediction, the maximum part of the dataset is used for training and the minimum part of the dataset is used for testing. Now, the predictions are done annually using the constructed Prophet model. Among the different attributes, ‘ds’, ’yhat’, ’yhat\_lower’, ’y\_hat \_ upper’ is only taken into consideration. The forecast of Prophet model is shown below



The daily, monthly and yearly prediction is shown below



**6 Conclusions**

The phenomenal demand for the natural gasses is the soul of this experimental research. One of the most signifying factors of natural gasses was that they are the most signifying factor for the conflict between Russia and Ukraine which was initially a business dispute that later made them declare a war among themselves. Natural gasses have a powerful impact over developed and developing nations.The variables of accuracy and stability are to be focused mainly in the area of forecasting. This experimental paper proposed models of ARIMA, LSTM, BLSTM and Neural Prophet in which the accuracy of individual models have been listed below