Capstone Project: Final Report

Project Title: Food Temperature Analysis and Forecasting

AIML 2019 (Hyderabad) Group #2

Abstract:

This report delves into the analysis of temperature time-series data, deployment of forecasting models and their effectiveness in predicting food temperature based on historical data. Temperature of several food items recorded over a period of three months has been utilized for this purpose.

Multiple Machine Learning models, and their effectiveness in predicting food temperature have been analysed. The results of these findings are discussed herein during the conclusion.

Keywords— Linear Regression, ARIMA, RNN, LSTM, Supervised Learning, Deep Learning, Time Series Plots, Sequential Models, Temperature Forecasting

Background & Objective:

Recent advances in Remote Sensing, Cloud Computation and Machine Learning hold the potential to revolutionize the domain of Food Safety. Now more than ever, it is possible to automate the maintenance of temperature, microorganism growth, humidity and other parameters pertaining to the upkeep of food standards in real time.

The goal is to build an algorithm that learns from historic temperature fluctuations in a given food sample and predicts the same for a specified time interval. Thus, the restaurant owner should be able to prevent or at least minimize losses due to depreciation of food quality.

Implementing Artificial Intelligence in Food Safety should yield greater food quality, lower operational costs and minimal losses from the spoilage of edibles. Such benefits would add tremendous value to the day to day operations of the Food Industry.

The effectiveness of the algorithms used is to be measured based on widely accepted metrics: Root Mean Squared Error, R^2, Mean Absolute Error and Mean Square Error. Should a model perform optimally as per prescribed food safety standards, it may be considered for real time application in the Food Industry.

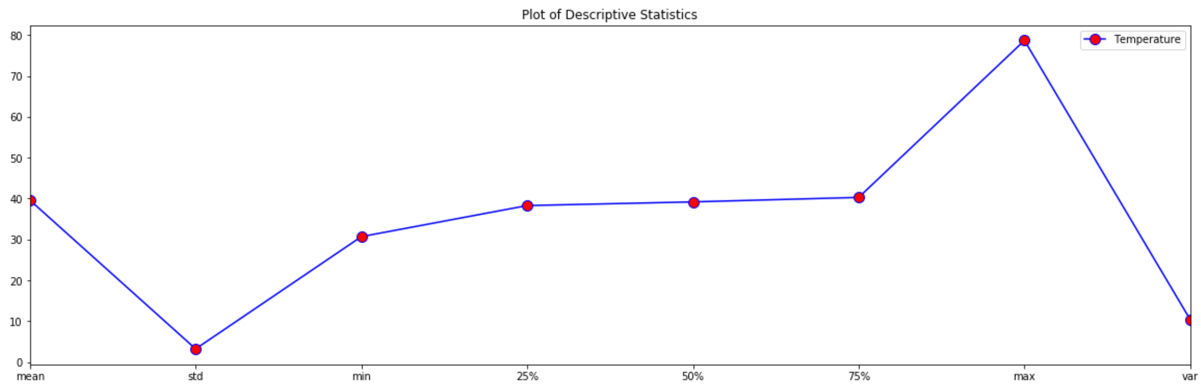
Methods:

The performance of the methods used to forecast temperature is gauged by its accuracy and the prediction interval length. Along the way, we also empirically demonstrate why classical benchmark statistical methods like ARIMA / (S)ARIMA are less preferable than LSTM. In the Conclusion, advantages, drawbacks and limitations of each model are highlighted.

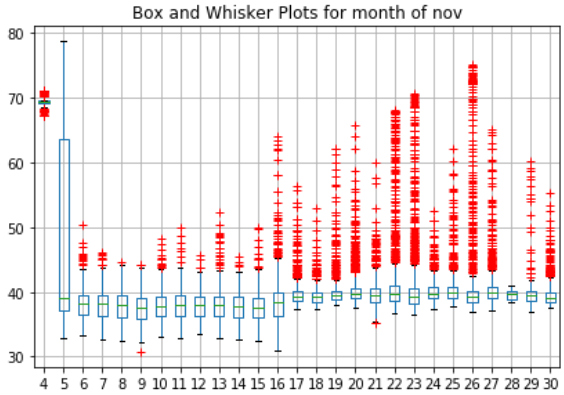
Temperature and Date Time readings are gathered continuously on a daily basis using IoT Sensors for a total duration of three months. Since the sensor readings are continuous, the 1-minute averages of said temperature readings along with their respective timestamps are gathered as individual data points and exported in csv format. Multiple sensors have been placed in different food samples within the restaurant's pantry. Each food sample is isolated from the other however they are all influenced by room temperature.

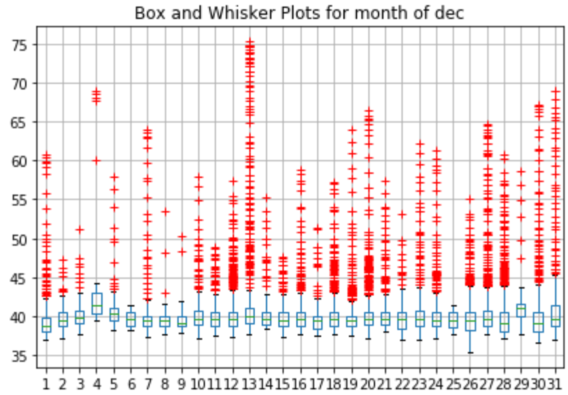
Data Visualization (Walk in Sensor)

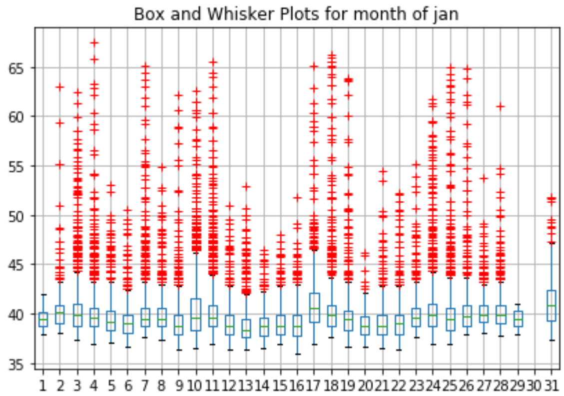
Data Summary:

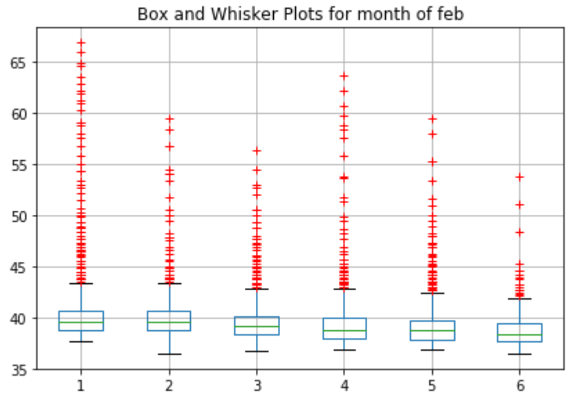


Box Plots:

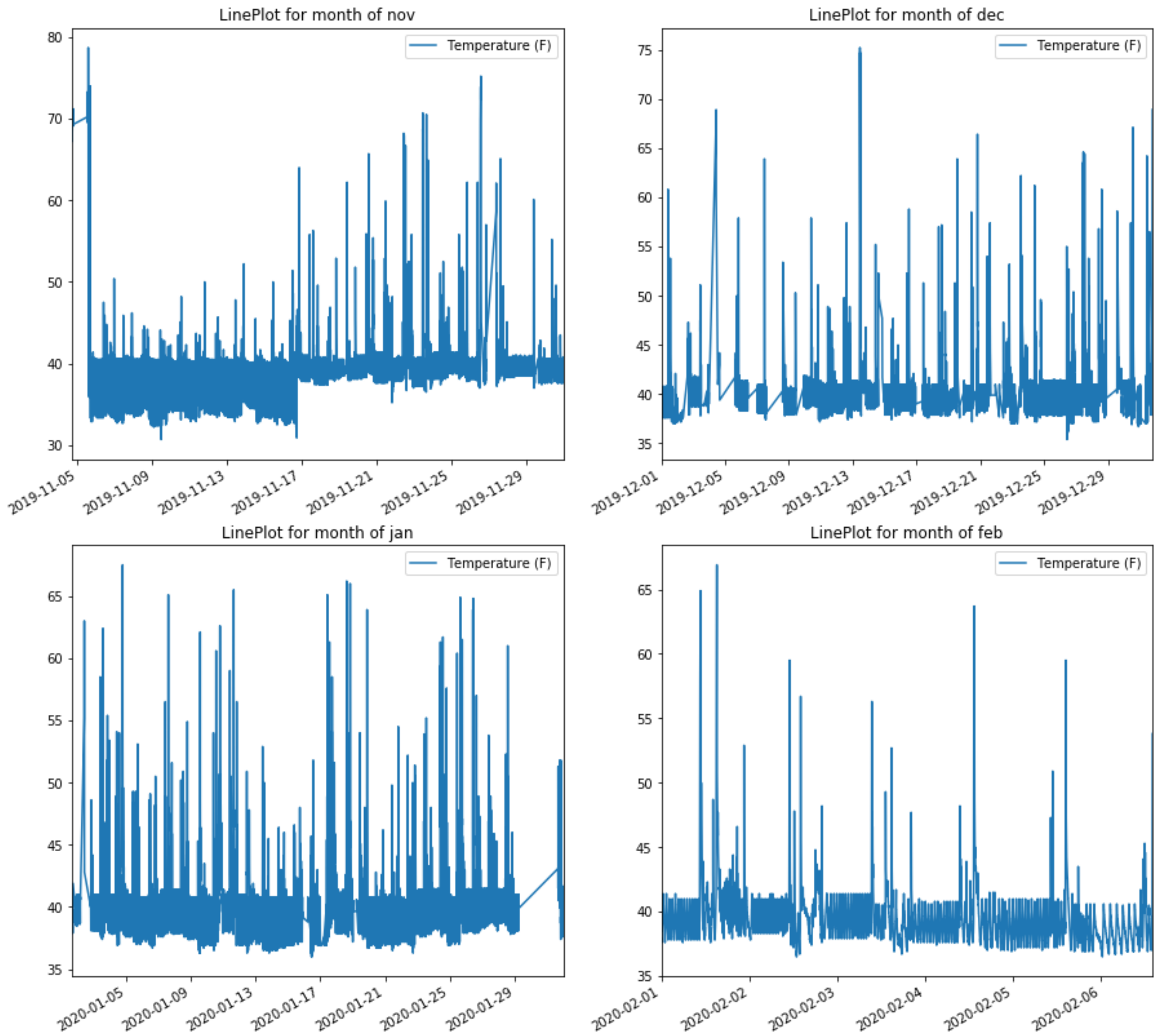

Box and Whisker Plots for month of nov
80
70
60
50
40
30
4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30


Box and Whisker Plots for month of dec
75
70
65
60
55
50
45
40
35
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

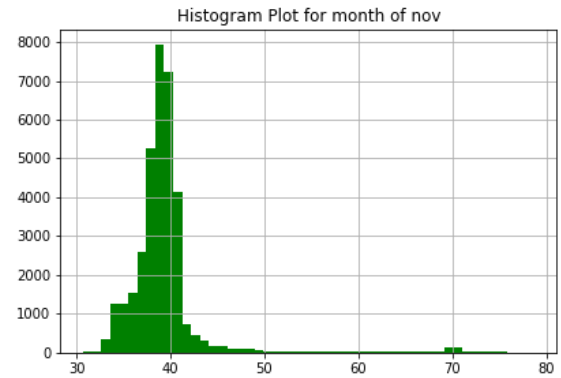

Box and Whisker Plots for month of jan
65
60
55
50
45
40
35
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

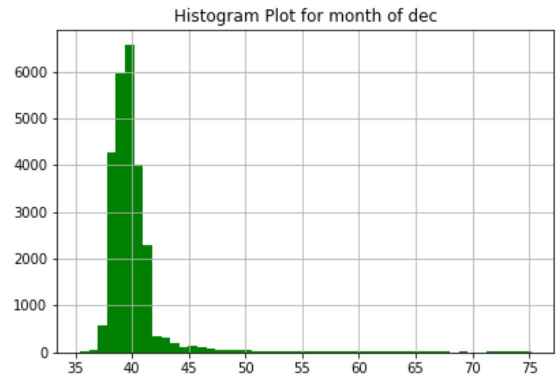

Box and Whisker Plots for month of feb
65
60
55
50
45
40
35
1 2 3 4 5 6

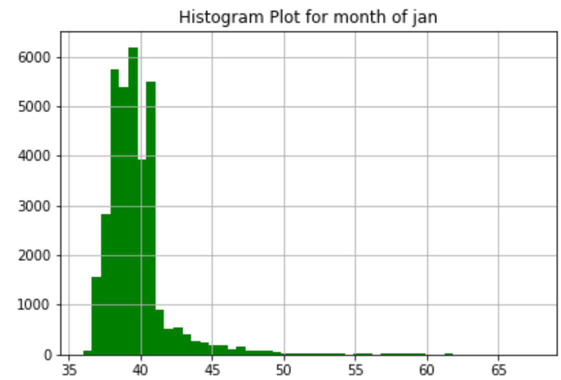
Line Plots:

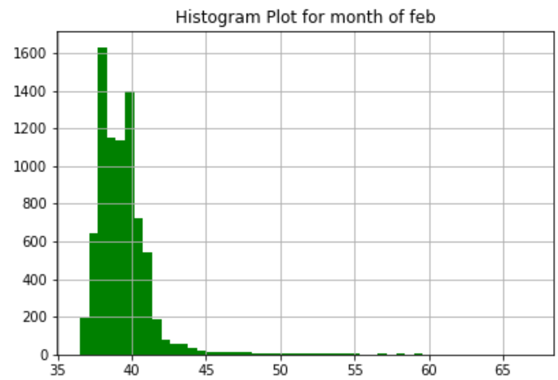

LinePlot for month of nov
Temperature (F)
LinePlot for month of dec
Temperature (F)
LinePlot for month of jan
Temperature (F)
LinePlot for month of feb
Temperature (F)

Histogram Plots:

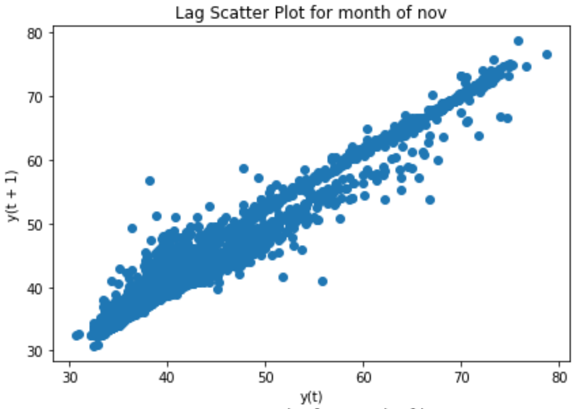


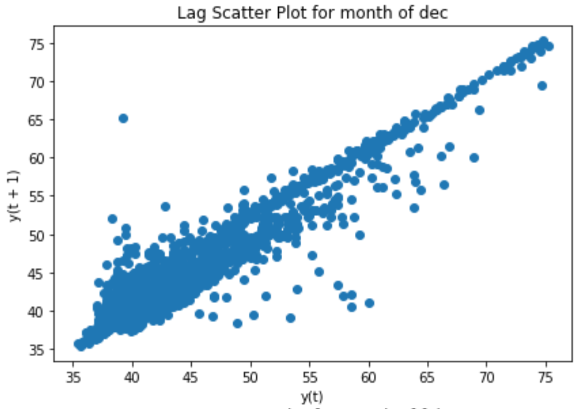


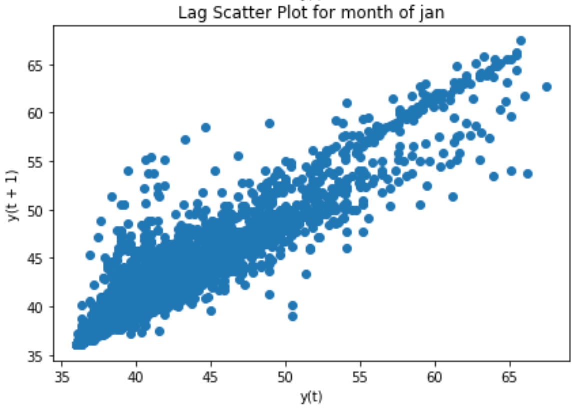


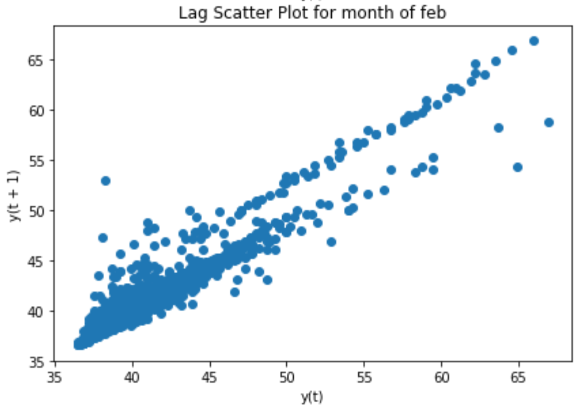


Lag Scatter Plots:







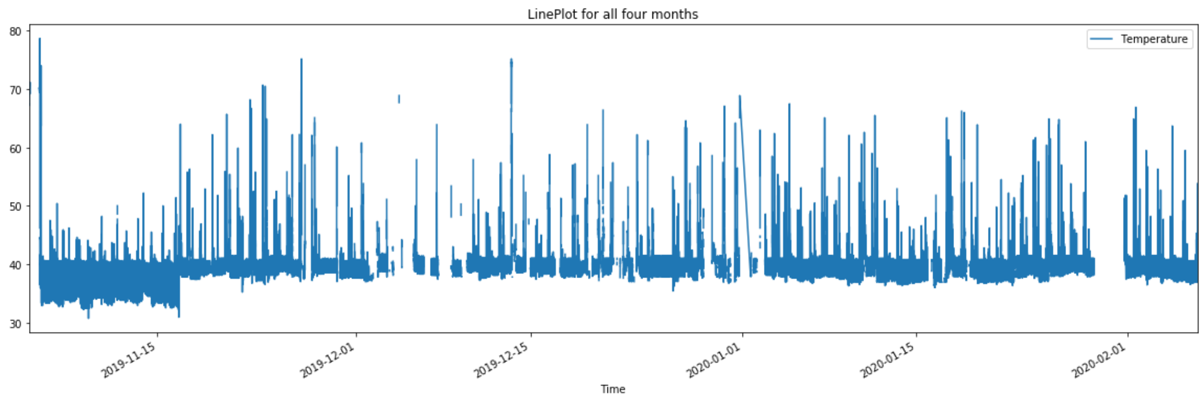


Observations:

- The sensor measurements are from early November to early February.
- There are a considerable number of outliers in each month's data. The constant opening and closing of refrigerator door caused the temperature to shoot up to around 70 ºF from the optimal 40 ºF.
- There are missing values that were not measured due to sensor issues. These values have not been imputed since the data available was sparse.
- The distribution is bell curve shaped with a right skew because most of the outliers are of high temperature.

Data Pre-Processing:

The data has been converted into a temperature, date and time format and then normalized. The outliers were left untreated because the rise and fall in temperature are genuine and occurred due to the opening and closing of pantry doors which caused the room temperature to briefly change.



Temperature Forecasting using Statistical Methods:

ARIMA:

ARIMA (Auto Regressive Integrated Moving Average) was conceptualized to predict a time series variable based on historic data. An autoregressive model (AR) operates on the notion that past values have an effect on current values. A Moving Average (MA) model assumes that the dependent variable of the current timestamp depends on the error terms of the previous timestamp. An ARMA (Auto Regressive Moving Average) is simply a combination of both of the aforementioned models, AR and MA.

The Box-Jenkins model also known as ARIMA, adds differencing to the ARMA model. Differencing subtracts the current value from the previous and can be used to transform a time series model into a stationary format. Three integers 'p' (number of autoregressive terms of AR order), 'd' (number of non-seasonal differences of differencing order) and 'q' (number of moving average terms of MA order) are typically used to parameterize ARIMA models.

The correlation between the observations at the current point in time and the observations at all previous points in time, known as Auto Correlation Factor (ACF) is used to determine the optimal Moving Average terms which in turn determines the order of the model. The Partial Auto Correlation Function (PACF) is a subset of ACF that expresses the correlation between observations made at two points in time while also factoring in influence from other data points. The PACF is also used to determine the order of the model.

Ljung Box Test (White Noise Test)

The Ljung Box Test is used to infer whether the series is white noise or not. We initiate the test with the Null Hypothesis being there's no autocorrelation between the signal and its lagged version. The Alternate Hypothesis to that is there is an autocorrelation between the signal and its lag.

Lag	Statistic	p-value
1	33341.472	0.000
2	65281.190	0.000
3	95571.570	0.000
4	124119.804	0.000
5	150895.514	0.000
6	175902.406	0.000
7	199186.114	0.000
8	220818.126	0.000
9	240892.619	0.000
10	259508.468	0.000
11	276772.555	0.000
12	292815.652	0.000
13	307776.292	0.000
14	321764.966	0.000
15	334898.554	0.000
16	347286.490	0.000
17	359023.912	0.000
18	370213.776	0.000
19	380945.594	0.000
20	391316.436	0.000
21	401421.332	0.000
22	411350.506	0.000
23	421175.313	0.000
24	430978.347	0.000
25	440825.161	0.000
26	450776.013	0.000
27	460872.031	0.000
28	471158.333	0.000
29	481653.641	0.000

Thus, we can infer that the given time series is not white noise as the p-value is close to 0 across several intervals. We reject the null hypothesis and state that the given series is not white noise.

Trend Stationarity Test (KPSS Test)

Testing the stationarity of the data is a critical step in predictive time series modelling. A stationary pattern is one who's mean and variance don't change with time. Trend stationarity is the pattern which, when detected and removed from the time series plot, results in a stationary plot. Doing so mitigates the effect of shocks on the pattern and visualise the prevalent underlying pattern.

The Trend Stationarity is tested by the KPSS Test. The Deterministic Trend is detected by the lack of a Random Component in the behaviour of the time series plot. The Null Hypothesis is that the data's stationary is deterministic. The Alternate Hypothesis is that the data's stationary is not deterministic.

KPSS Statistic	3.637
p-value	0.01
Critical Value (1%)	0.739
Critical Value (2.5%)	0.574
Critical Value (5%)	0.463
Critical Value (10%)	0.347

Since the p value is below 0.05, we reject the null hypothesis and state that the type of Stationarity is not Trend Stationarity. This poses a challenge in terms of model building since the absence of a strong underlying pattern makes prediction harder.

Unit Root Test (Augmented Dickens-Fuller)

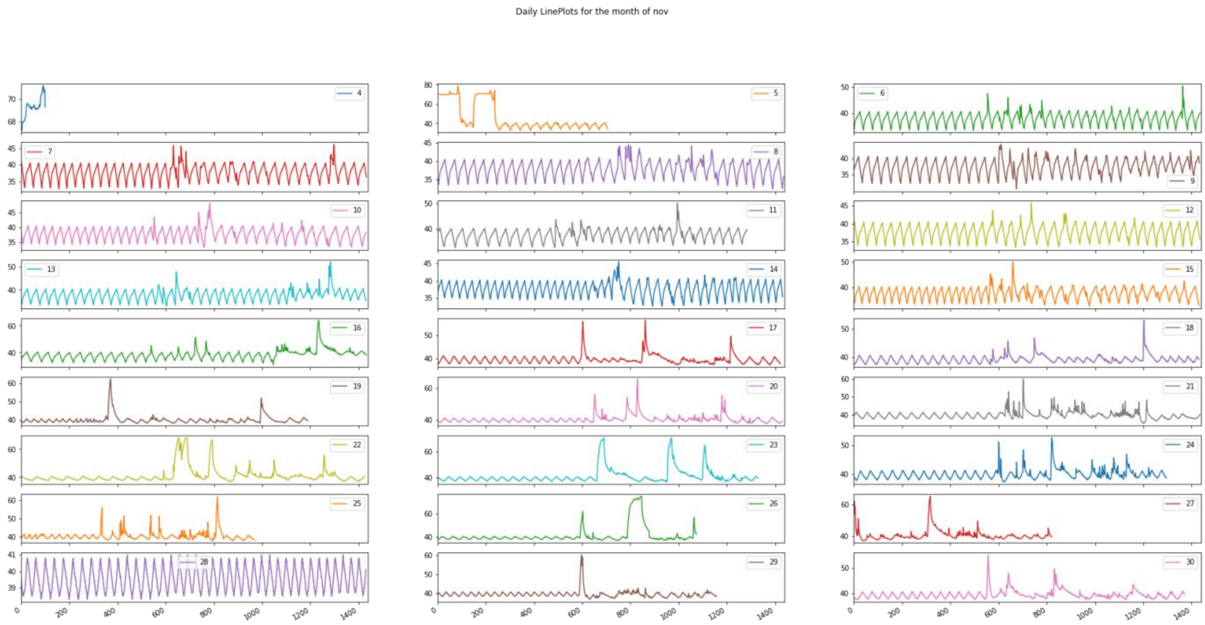
The Stochastic Stationarity in the data is determined by its Unit Root. In order to check whether the temperature time series has a unit root or not, we employ the Augmented Dickens Fuller Test. The Null Hypothesis is that the data has a unit root and does not have Stochastic Stationarity. The Alternate Hypothesis is that the data doesn't have a Unit Root and exhibits Stochastic Stationary.

ADF Statistic	-30.49
p-value	0.00
Critical Value (1%)	-3.430
Critical Value (5%)	-2.861
Critical Value (10%)	-2.567

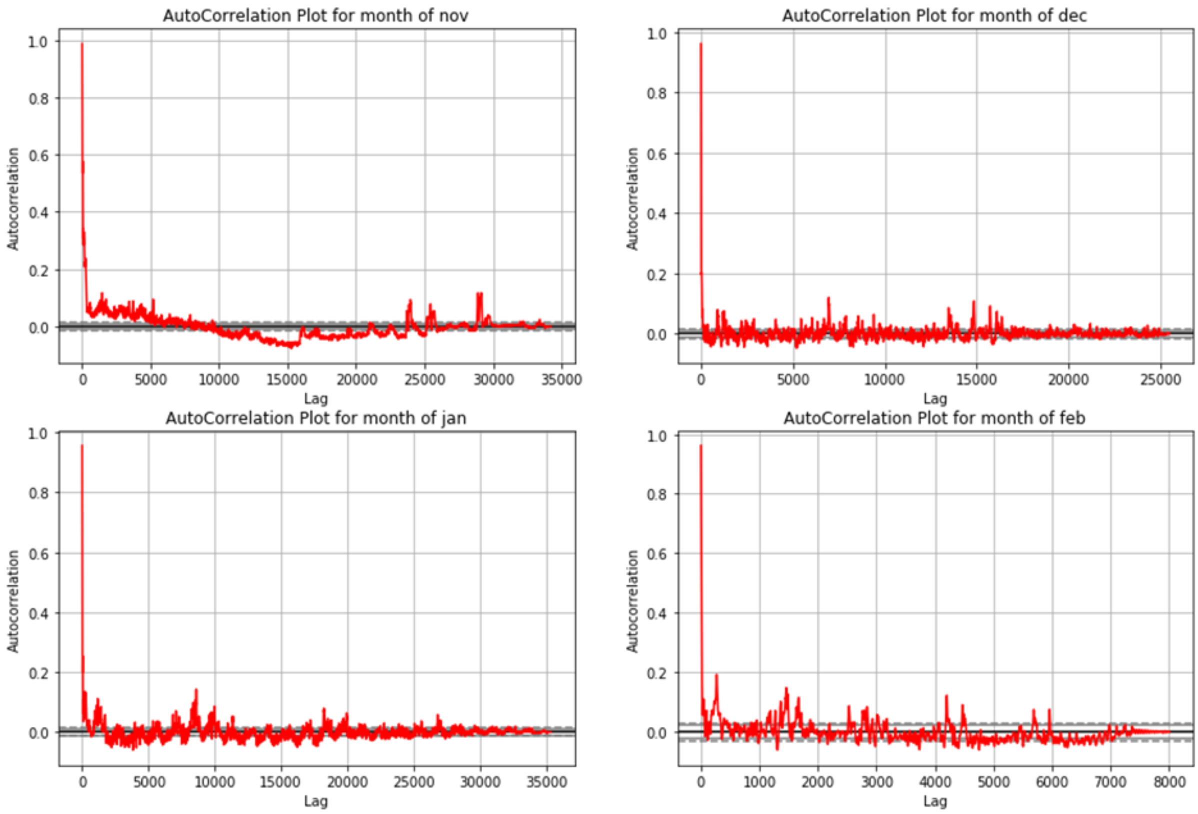
As the above p-value < 0.05, we reject the null hypothesis i.e. we may consider that the data has Stochastic Stationarity.

Besides *Stationarity, Seasonality* and *Trend* of the data are to be addressed prior to model building. By visualizing the data, we observe that there is no clear trend in the data. Seasonality occurs when the Time Series data is affected by seasonal factors like Time of the Year, Day of the Week in a fixed calendar frequency. Autocorrelation plots and PACF Plots are used to determine seasonality.

Seasonality Plots

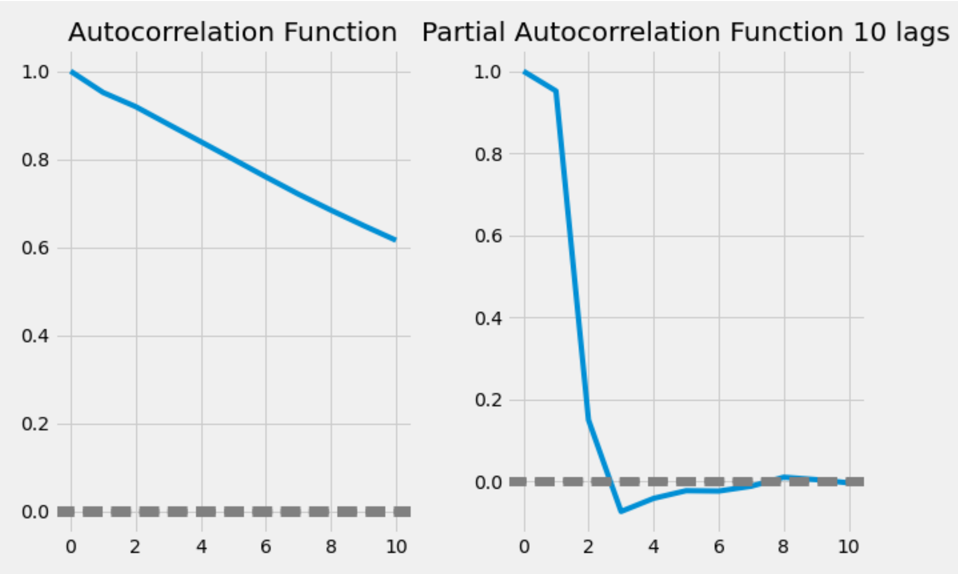


Autocorrelation Plots



ACF and PACF Plots

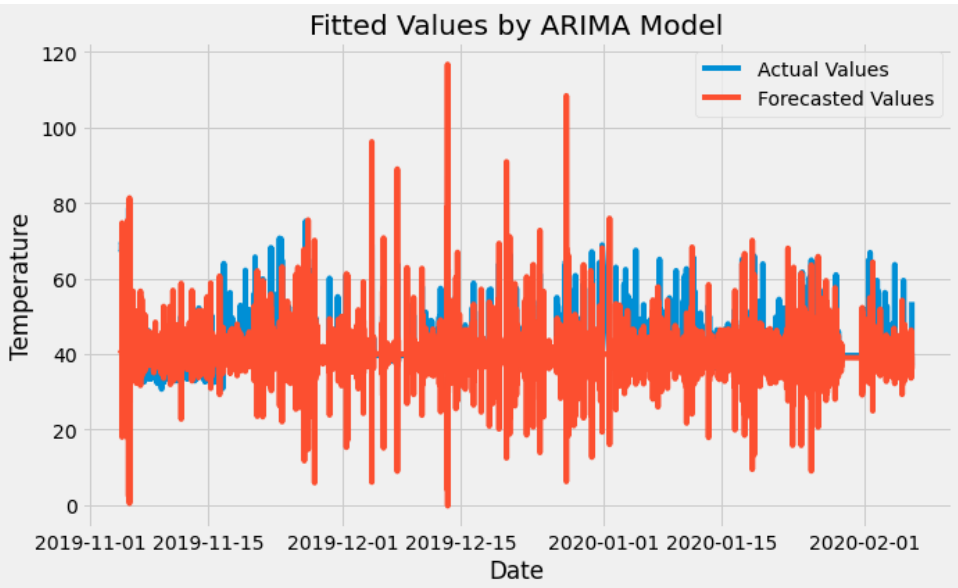
The autocorrelation plots for each of the months shows that the ACF sharply declines initially and thereafter, it hovers around 0.

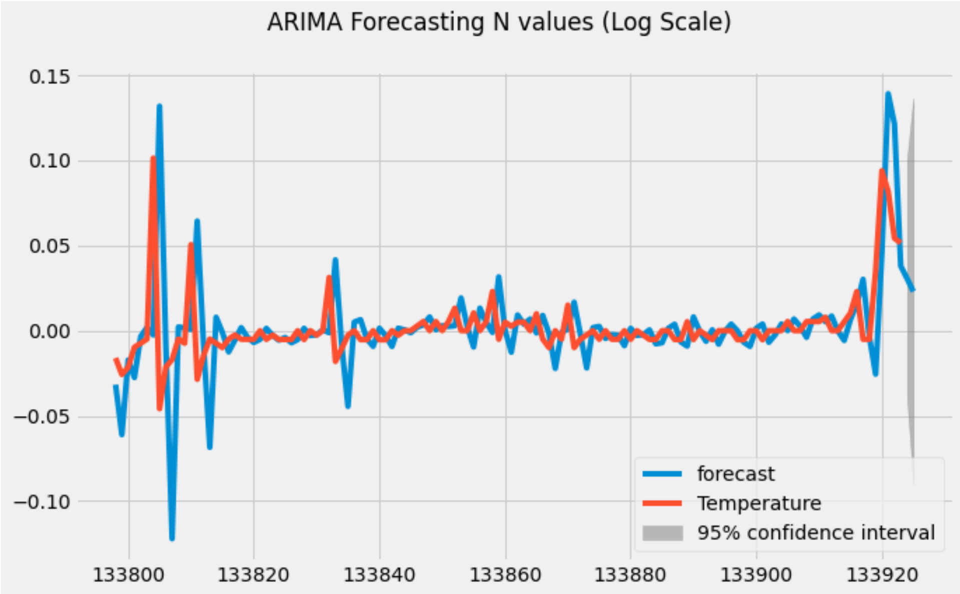


When 10 lags are considered, PACF drops after 2 lags as shown above. Thus, we conclude that there is no seasonality between values after 4-5 subsequent data points. These ACF and PACF plots show that seasonality is between 0-2.

Results of ARIMA Forecast

Using grid search we find the optimal (p=1,d=2,q=0) parameters for the implementation of ARIMA. We measure and compare the prediction against the actual temperature values.





ARIMA Metrics for Walk In Sensor

Mean Absolute Percentage Error for fitted values (MAPE): 0.87

Mean Squared Error for fitted values (MSE): 0.38

Duration	1 min	5 min	10 min	15 min	30 min	60 min	120 min	180 min
MAPE (Forecast)	1.42	0.69	1.99	9.43	6.71	3.39	3.33	4.46
RMSE (Forecast)	0.62	0.37	0.95	4.33	3.37	1.95	1.56	1.91

ARIMA Metrics for the entire dataset

ARIMA RESULTS			
Train MSE	Train RMSE	MAPE	R-Square
0.52	0.72	0.868798585	0.986676116

ARIMA Metrics time step breakdown

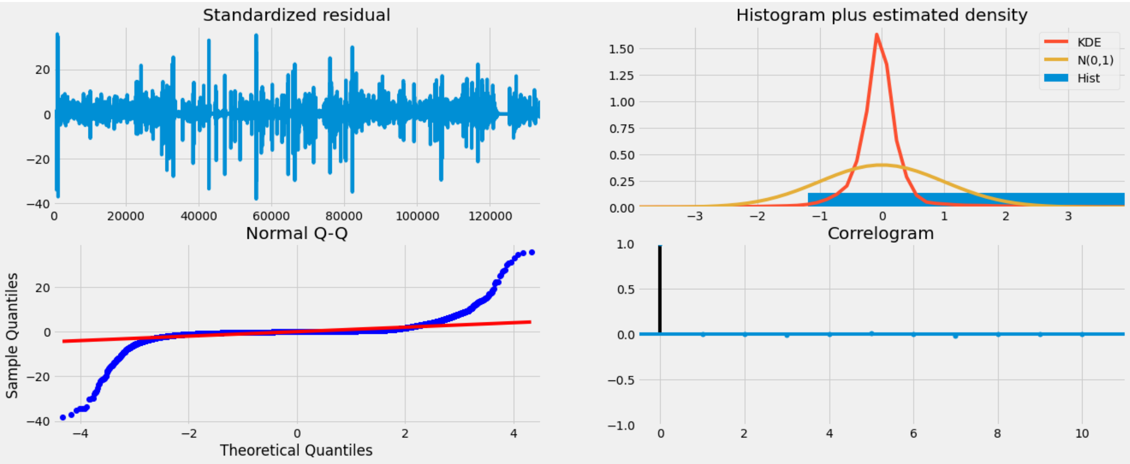
ARIMA RESULTS			
Forecasting Next	Test MSE	Test RMSE	Test MAPE
1	0.38	0.62	1.42602622
5	0.14	0.37	0.669931299
10	0.89	0.95	1.992238442
15	18.72	4.33	9.438480259
30	11.35	3.37	6.717362951
60	3.8	1.95	3.389628978
90	3.82	1.95	4.624790902
120	2.42	1.56	3.338980725
150	2.39	1.54	3.408343084
180	3.65	1.91	4.464382412

(S)ARIMA

Seasonal ARIMA, is an extension of ARIMA that explicitly supports univariate time series data with a seasonal component. SARIMA is used for non-stationary series, that is, where the data do not fluctuate around the same mean, variance and covariance. We can see that there is a spike in ACF and PACF at 1 and 2. We use grid search for finding the optimal (p,d,q)(P,D,Q)1 and (p,d,q)(P,D,Q)2 with seasonality 1 and 2 respectively. We use Akaike Information Criterion and Bayesian Information Criterion for choosing the most optimal model for our data. We end up with the parameters (3,0,1)(3,1,1,1) as out optimal parameters from grid search.

Seasonal ARIMA is formed by including additional seasonal terms. The parameters for seasonal ARIMA are [(p,d,q), (P,D,Q) ,m]. The seasonal part of AR or MA will be present in the seasonal lags of PACF and ACF.

Model	Coefficient	Standard Error	Z	P > |Z|	[0.025	0.975]
Ar.L1	-0.0439	0.03	-1.455	0.146	-0.103	0.015
Ar.L2	0.0352	0.002	14.255	0.000	0.03	0.04
Ar.L3	0.0067	0.001	6.912	0.000	0.005	0.009
Ma.L1	-0.0373	0.03	-1.235	0.217	-0.096	0.022
Ar.S.L1	-0.0439	0.03	-1.455	0.146	-0.103	0.015
Ar.S.L2	0.0352	0.002	14.255	0.000	0.03	0.04
Ar.S.L3	0.0067	0.001	6.912	0.000	0.005	0.009
Ma.S.L1	-0.0373	0.03	-1.235	0.217	-0.096	0.022
sigma2	0.7428	0.00	2669.466	0.000	0.742	0.743

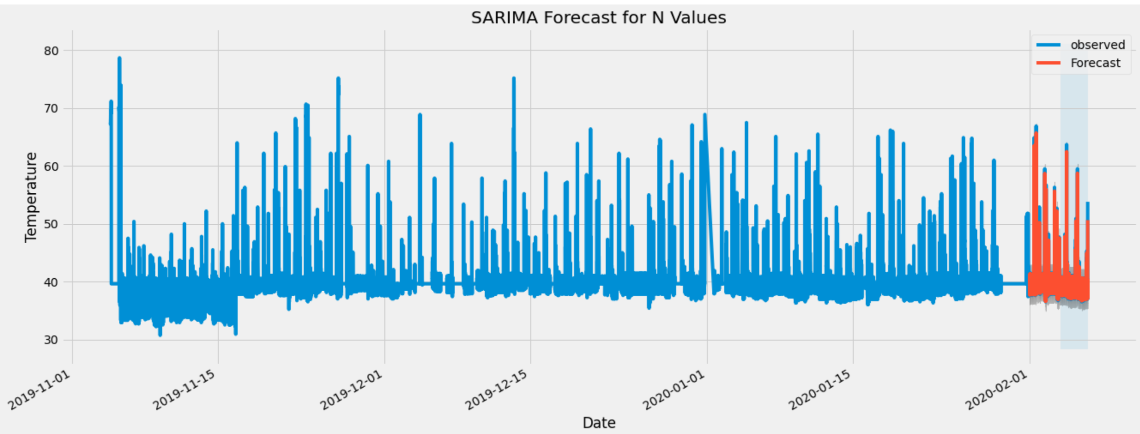


1) The KDE plot follows a normal distribution with mean zero. A good indication that the residuals are normally distributed.

2) The qq-plot on the bottom left shows that the ordered distribution of residuals follows the non-linear trend of the samples taken from a standard normal distribution with N(0, 1).

3) The residuals over time don't display any obvious seasonality and appear to be white noise.

4) This is confirmed by the autocorrelation plot on the bottom right, which shows that the time series residuals have low correlation with lagged versions of itself.



MAPE (Fitted values) = 0.726

MSE (Fitted values) = 0.776

Duration	1 min	5 min	10 min	15 min	30 min	60 min	120 min	180 min
MAPE (Forecast)	5.26	16.94	10.84	8.84	4.62	5.35	5.30	3.72
RMSE (Forecast)	2.83	9.68	7.58	6.62	4.59	3.18	2.74	2.39

Advantages of ARIMA:

1) Very effective for modelling stationary data or data with constant mean and variance.
2) Very effective is there is a repeated pattern in Trend, Seasonality or Cyclicity.
3) Traditional Model and easy to interpret results.
4) Good for modelling time series data where there is an underlying pattern between present and past data.

Disadvantages of ARIMA:

1) It relies on the assumption that future predictions are dependent on historic data.
2) Not easy to model when there is lack of stationarity or seasonality.
3) Does not consider other external factors.

Forecasting Using Classical Machine Learning Techniques

Linear Regression

Linear Regression creates a line of best fit between Time and Temperature for the given data. It is the most commonly used algorithm for solving regression problems.

KNN Regressor

KNN algorithm forecasts the unknown temperature data point by aggregating its value from its nearest known data points. There are several methods to calculate this distance based on which the prediction depends.

Support Vector Machine Regressor

SVM Regressor creates a hyperplane that fits the most number of points from the training set and then uses the same to predict the test dataset points.

Decision Tree Regressor

The Decision Tree algorithm implements a top down greedy search creating branching possibilities and nodes along the way. This creates a tree structure that branches into smaller subsets on the basis of standard deviation reduction.

Random Forest Regressor

Random Forest is an ensemble technique that involves creating multiple decision trees and deriving the average of their structure to create a more generalizable version of the Decision Tree Algorithm.

Forecasting using Deep Neural Networks (LSTM: Long Short-Term Memory)

Recurrent Neural Networks (RNN) are suited for Time Series forecasting since they are capable of learning from historic data multiple times to predict future data points. An input into an RNN Cell returns a state which is then used as an input to next cell in the chain of RNNs. This process continues until the entire time series sequence has been trained and the information from all previous states has been joined.

$$S_t = f(UX_t + WS_{t-1}) \quad O_t = softmax(VS_t)$$

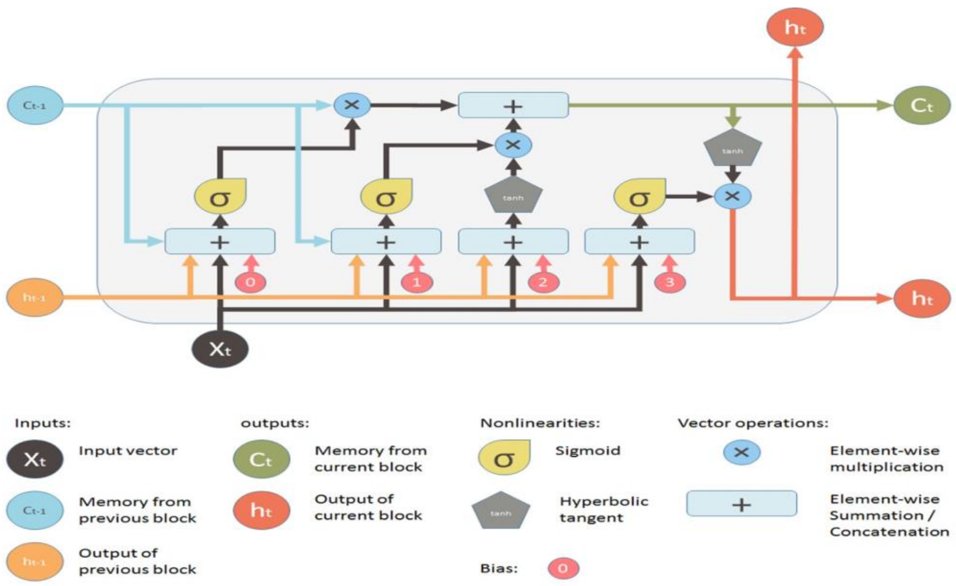
The error resulting from the final state is measured against the actual value gets registered as Loss. This loss is backpropagated and optimized thus updating the weights and training the model.

$$\frac{dLoss}{dW} = \frac{dLoss}{dO_{t+3}} * \frac{dO_{t+3}}{dS_{t+3}} * \frac{dS_{t+3}}{dS_{t+2}} * \frac{dS_{t+2}}{dS_{t+1}} * \frac{dS_{t+1}}{dS_t} * \frac{dS_t}{dW}$$

But the major disadvantage of RNNs is that they cannot process very long sequences due to Vanishing Gradient / Exploding Gradient during the training process caused by greater lags.

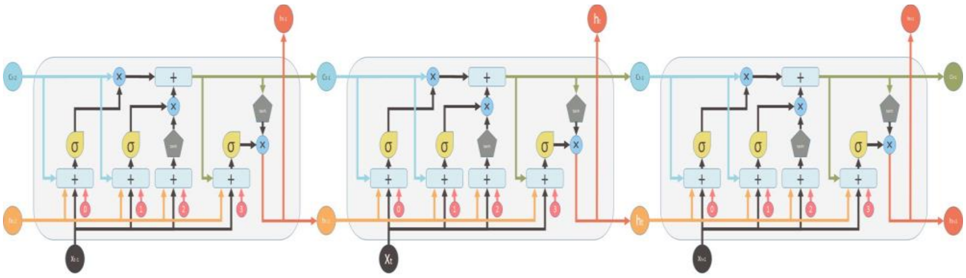
LSTMs were specifically designed as modified RNNs to avoid long-term dependency which makes it easier to remember past data in its memory.

LSTM Cell Architecture



An LSTM Cell consists of three gates i.e. Input Gate which adds to Cell State, the Forget Gate which controls what gets retained in the Cell State and finally, the Output Gate which controls the information stored in the hidden state.

Multiple Neural Networks in LSTM



LSTMs also have a chain structure like RNNs; however, the repeating module has a different structure. Instead of having a single neural network layer, there are four, interacting in a special manner.

$$S_{t+n} = tanh(WS_{t+n-1} + UX_{t+n})$$

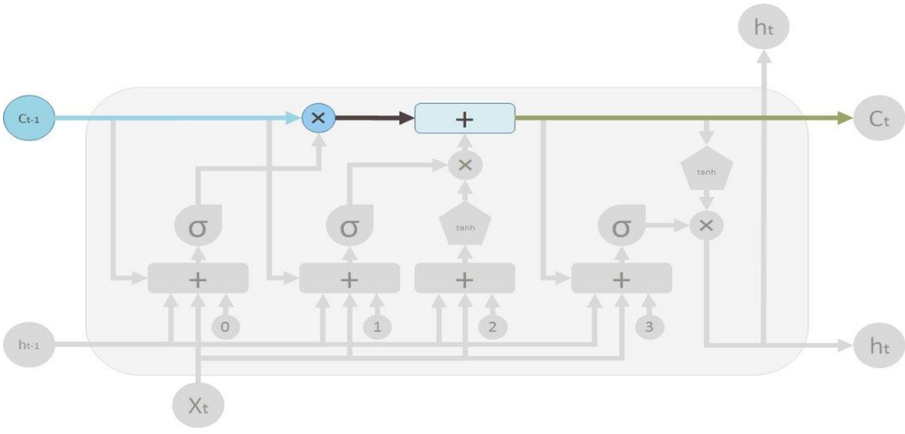
$$\frac{dLoss}{dW} = \frac{dLoss}{dO_{t+3}} * \frac{dO_{t+3}}{dS_{t+3}} * \frac{dS_{t+3}}{dS_{t+2}} * \frac{dS_{t+2}}{dS_{t+1}} * \frac{dS_{t+1}}{dS_t} * \frac{dS_t}{dW}$$

$$\frac{dS_{t+n}}{dS_{t+n-1}} = W$$

$$\frac{dLoss}{dW} = \frac{dLoss}{dO_{t+3}} * \frac{dO_{t+3}}{dS_{t+3}} * W * W * W * \frac{dS_t}{dW}$$

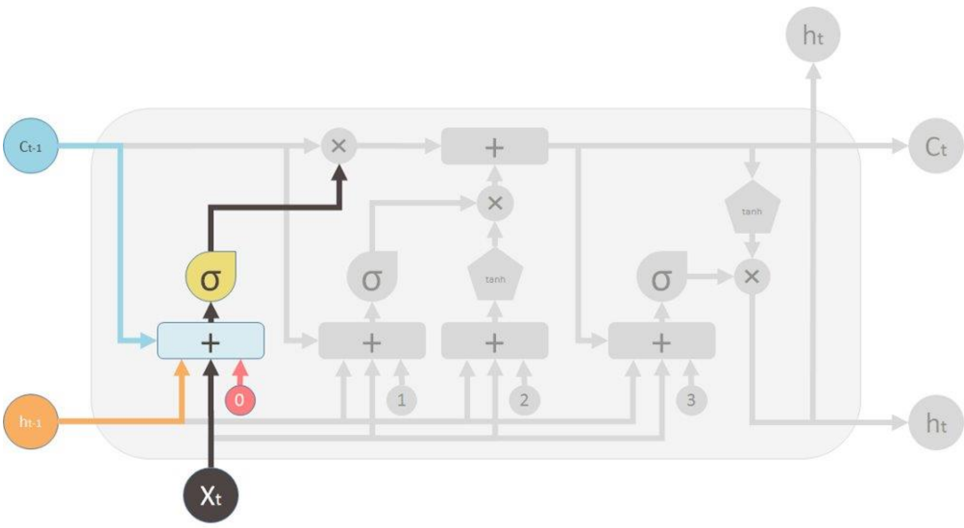
Cell State

An LSTM Cell has two states: A Cell State and a Hidden State. The Cell State stores long term memory and with the onset of fresh input, older information gets retained or removed based on the workings of the Input Gate and Forget Gate.



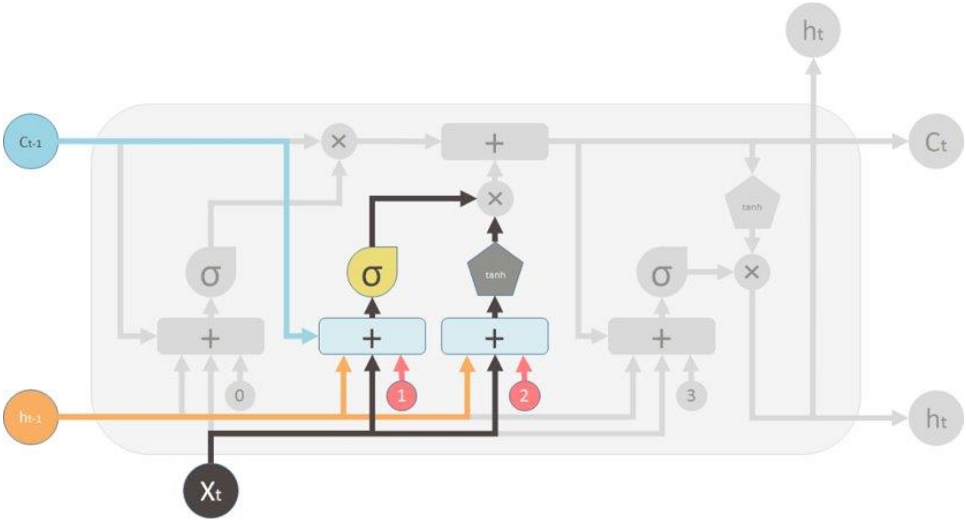
$$i_t = \sigma(w_i[h_{t-1}, x_t] + b_i)$$
$$f_t = \sigma(w_f[h_{t-1}, x_t] + b_f)$$
$$o_t = \sigma(w_o[h_{t-1}, x_t] + b_o)$$

LSTM Forget Gate



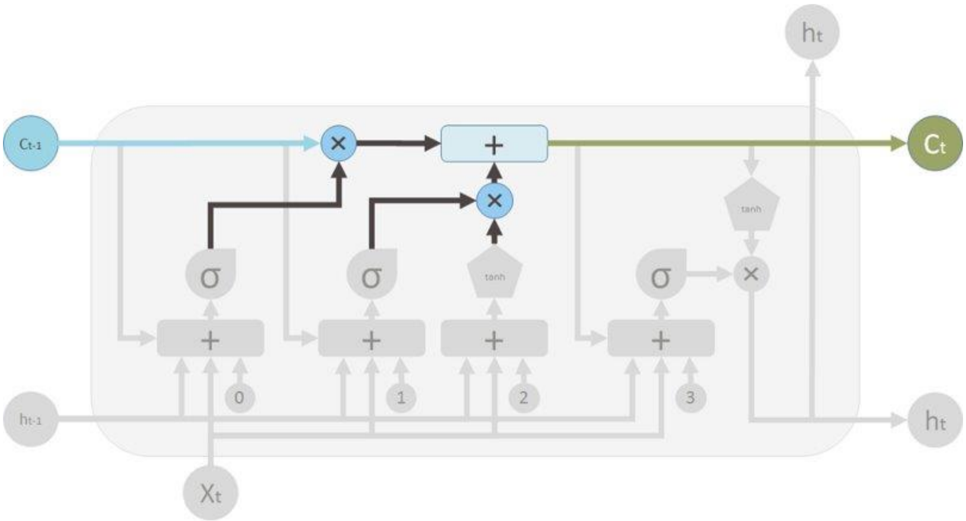
The range of the Forget gate's output is between 0 and 1 due to its sigmoid function. The size of the returned value from the forget gate has the same size as the Cell State enabling element wise multiplication between the two of them.

Input Gate



A Cell state is derived from its local memory at a particular time. Input gates are responsible for the passage of elements from Cell State to Long Term Cell State. Hence the Input Gate and the Cell State have the same size.

Updated Cell

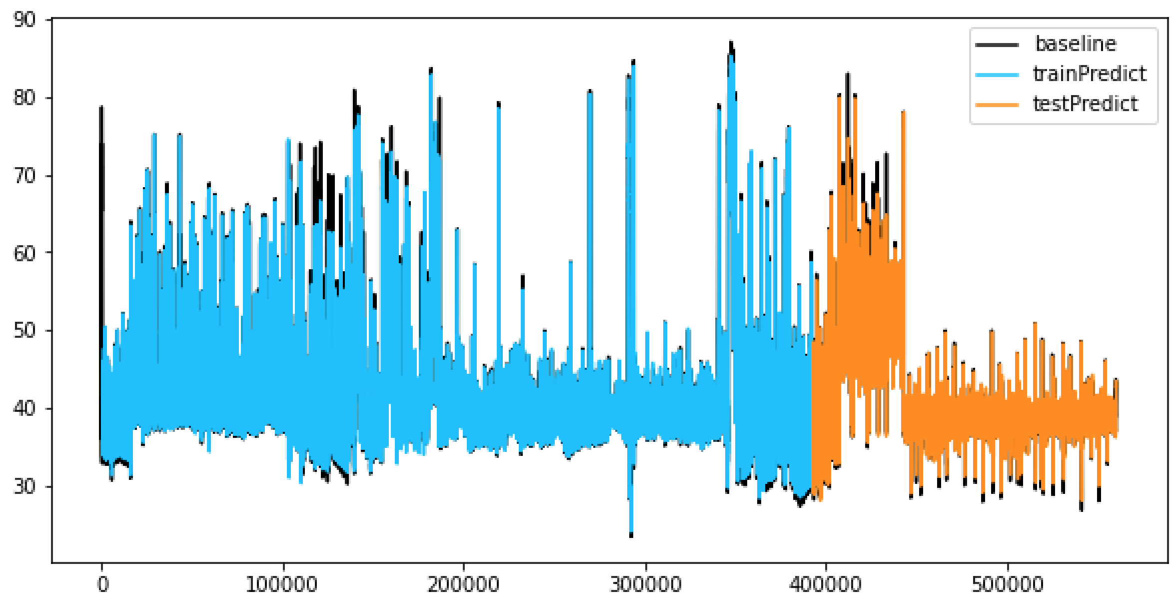


The Output gate is used to control the flow of information from Cell State to Hidden State. The hidden state has the same size as Cell State and is used to make the final prediction in real time situations. The prediction can be compared against actual data to determine the metrics of the algorithm.

Results

Window Size	Model Name	Train MAE	Test MAE	Train MSE	Test MSE	Train RMSE	Test RMSE	Train R2 Score	Test R2 Score
60	LR	0.0562	0.0494	0.0143	0.0088	0.1195	0.0936	0.9878	0.9853
60	KNN	0.0475	0.1103	0.0119	0.0289	0.1091	0.1701	0.9898	0.9516
60	SVM	0.0515	0.0485	0.0125	0.0082	0.1118	0.0908	0.9893	0.9862
60	DT	0.047	0.0554	0.0103	0.0098	0.1016	0.0992	0.9912	0.9835
60	RF	0.045	0.0536	0.0098	0.0093	0.0988	0.0962	0.9917	0.9845
60	LSTM	0.0585	0.0575	0.0139	0.0093	0.1178	0.0964	0.9881	0.9845
60	RNN	0.0549	0.0523	0.015	0.0095	0.1224	0.0974	0.9872	0.9841
60	GRU	0.0563	0.0519	0.0134	0.0084	0.1159	0.0919	0.9885	0.9859
120	LR	0.0562	0.0495	0.0143	0.0088	0.1194	0.0936	0.9878	0.9853
120	KNN	0.0482	0.1451	0.0115	0.0489	0.107	0.2211	0.9902	0.9182
120	SVM	0.0527	0.0504	0.0128	0.0089	0.1132	0.0942	0.989	0.9852
120	DT	0.0472	0.0559	0.0103	0.0111	0.1012	0.1056	0.9912	0.9813
120	RF	0.0445	0.0533	0.0093	0.0091	0.0964	0.0954	0.992	0.9848
120	LSTM	0.0545	0.0482	0.0138	0.0085	0.1174	0.0921	0.9882	0.9858
120	RNN	0.0667	0.061	0.0157	0.0101	0.1253	0.1006	0.9865	0.9831
120	GRU	0.0534	0.0511	0.0135	0.0089	0.1161	0.0941	0.9884	0.9852
180	LR	0.0562	0.0495	0.0142	0.0088	0.1193	0.0936	0.9878	0.9853
180	KNN	0.047	0.1693	0.0107	0.067	0.1034	0.2588	0.9908	0.8879
180	SVM*	Did Not Run	Did Not Run	Did Not Run	Did Not Run	Did Not Run	Did Not Run	Did Not Run	Did Not Run
180	DT*	Did Not Run	Did Not Run	Did Not Run	Did Not Run	Did Not Run	Did Not Run	Did Not Run	Did Not Run
180	RF	0.0445	0.0533	0.0092	0.0091	0.096	0.0955	0.9921	0.9847
180	LSTM	0.0579	0.0523	0.014	0.0088	0.1182	0.0938	0.988	0.9853
180	RNN	0.0557	0.0578	0.0145	0.0098	0.1204	0.0988	0.9875	0.9837
180	GRU	0.056	0.0521	0.0139	0.0087	0.1178	0.0934	0.9881	0.9854

LSTM Prediction for 600 Minutes



Conclusion

As can be observed from the visualization and tables above, the Machine Learning and Deep Learning models are superseding the benchmark ARIMA model both in terms of prediction accuracy and time complexity and thus have been given more preference.

Of the given algorithms, SVM is taking unusually long during the training phase. Though CART algorithms are performing well in training, they don’t generalize well in real time.

While Linear Regression is performing on par with LSTM, LSTM is better at predicting both outliers and central values simultaneously. This is important because the preservation of food quality relies on being able to predict suboptimal outlier temperatures that are too hot or cold. Hence our algorithm of choice for solving this problem is LSTM.

Implications and Limitations

Since the data available was sparse, any further model improvement would require additional data to be gathered. The predictions of the models used in the project would then have to be compared against new data to fully measure their worth. More temperature data, from ideally the same environment as the last dataset, should give better predictions and be able to add value to food safety by offering a preventive prediction.

Literature Survey:

1.) Meiyu Wen, Dandan Che, Jean-Pierre Niyigena, Ruohan Li, Qingshan Jiang, 2019 "A Prediction Model of IoT Data Using Long Short-Term Memory Neural Network" - ISAICN

2.) Alexandru Popa, Mihaela Hnatiuc, Mirel Paun, Oana Geman, D. Jude Hemanth, Daniel Dorcea, Le Hoang Son and Simona Ghita ,2019 "An Intelligent IoT-Based Food Quality Monitoring Approach Using Low-Cost Sensors" - symmetry

3.) Roberto Casado-Vara, Paulo Novais, Ana Belen, Javier Prieto, Juan Manuel Corchado, 2018 "Distributed Continuous-Time Fault Estimation Control for Multiple Devices in IoT Networks"- IEEE

4.) V.Sandeep, K.Lalith Gopal, S.Naveen, A.Amudhan, L. S. Kumar, 2015 "Globally Accessible Machine Automation Using Raspberry Pi Based on Internet of Things" - ResearchGate

5.) Pedro Lima Monteiro, Massimiliano Zanin, Ernestina Menasalvas Ruiz, João Pimentão, Pedro Alexandre da Costa Sousa, 2018 "Indoor Temperature Prediction in an IoT Scenario" - sensors

6.) M.K. Nallakaruppan, U. Senthil Kumaran, 2019 "IoT based Machine Learning Techniques for Climate Predictive Analysis" - IJRTE

7.) Marco Ferrari, Msrco Bau, Vittorio Ferrari, 2014 "Resonant Piezo-layer (RPL) Sensors with Contactless Interrogation for Food Monitoring from Outside Sealed Packages" - ScienceDirect

8.) A. Flammini, E. Sisinni, 2014 "Wireless Sensor Networking in the Internet of Things and Cloud Computing Era" - ScienceDirect

9.) Ganesan Sangeetha, Muthuswamy Vijayalakshmi, "Role of Smart Sensors in Minimizing Food Deficit by Prediction of Shelf-Life in Agricultural Supply Chain"

10.) BEN ABDEL OUAHAB IKRAM, BOUDHIR ANOUAR ABDELHAKIM, ASTITO ABDELALI, BASSAM ZAFAR, BOUHORMA MOHAMMED, 2019 "Deep Learning architecture for temperature forecasting in an IoT LoRa based system" - ResearchGate

11.) Ganjar Alfiana, Muhammad Syafrudinb, Umar Farooqc, Muhammad Rifqi Ma'arifd, M. Alex Syaekhonib, Norma Latif Fitriyanib, Jaeho Leea, Jongtae Rheeb, 2020 “Improving efficiency of RFID-based traceability system for perishable food by utilizing IoT sensors and machine learning model” - Food Control (Elsevier)

12.) Nevena Golubovic, Rich Wolski, Chandra Krintz, Santa Barbara, Markus Mock, 2019 "Improving the Accuracy of Outdoor Temperature Prediction by IoT Devices" - IEE International Congress on IoT

13.) Hairong Yan, Michele Luvisotto, Kai Lu, Siyu Jiang, 2019 "Temperature prediction intelligent system based on BP neural network in wireless industrial IoT" - IEE Conference on Smart IoT

14.) Siddhartha Bhandari, Neil Bergmann, Raja Jurdak, Branislav Kusy,2017 "Time Series Analysis for Spatial Node Selection in Environment Monitoring Sensor Networks" - sensors

15.) Melanie Swan, 2012 "Sensor Mania! The Internet of Things, Wearable Computing, Objective Metrics, and the Quantified Self 2.0 - JSAN

16.) Radu Manuca, Robert Savit, 1996 "Stationarity and nonstationarity in time series analysis" - Physica D

17) Principles of Internet of Things (IoT) Ecosystem: Insight Paradigm by Sheng-Lung Peng, Souvik Pal, Lianfen Huang (pp 153-175) - Springer (Vol. 174)