MAKERERE UNIVERSITY

DESIGN OF A MACHINE LEARNING BASED SYSTEM FOR PHARMACEUTICAL PURCHASES

Abraham Jerry Kakooza
Reg No.: 16/U/327

Department of Electrical and Computer Engineering
School of Engineering
College of Engineering, Design, Art and Technology

Supervisor: **Dr Andrew Katumba**
Co-supervisor: **Prof. Eng. Dr. Peter Lating, iLabs Principal Investigator**

December 7, 2020

A Report submitted in partial fulfillment of the requirements for the
Degree of Bachelor of Science in Telecommunications Engineering
at Makerere University.

Declaration

ABRAHAM KAKOOZA JERRY

Academic Integrity Pledge:

I HAVE ABIDED BY THE MAKERERE UNIVERSITY ACADEMIC INTEGRITY POLICY ON THIS ASSIGNMENT. I also confirm that this report is only prepared for my academic requirement, not for any other purpose.

Signature ____________________ *Date* ____________________

Approval

This report has been submitted with approval and under the supervision of the following supervisors.

Supervisor

Dr.Andrew Katumba ____________________ *Date* ____________________

Co-Supervisor

Prof.Peter Lating ____________________ *Date* ____________________

Dedication

I dedicate this to my parents who have been there for me in this entire study period both emotionally and financially, my supervisor Dr. Andrew Katumba, my mentors Tibabwetiza Joel, my partner Kevin Upyem, and lastly the entire ilabs team Mark Phillip and Ronald Ogwang .

Acknowledgments

First of all, I would like to thank God for enabling me to complete my Final Year Project successfully. I also extend my gratitude ti the ilabs at Mak Project for the opportunity to work with them which has given me a wealth of skills and experience to build my engineering career.

I highly acknowledge my main supervisor Dr. Andrew Katumba and my co-supervisor Prof. Peter Lating for the support, guidance and knowledge provided at each step of this project. I highly acknowledge my mentors David Tusubira and Tibabwetiza Joel for the guidance and support.

Lastly I would like to thank the Soteria Pharmacy community, Ms. Brenda , Ms.Lydia Ssebunya and the ilabs team Mark Phillip and Ogwang Ronnie for the collective support.

Abstract

To obtain inherent laws from vast amounts of pharmaceutical sales data and to provide valuable information to pharmacy managers, this work validates different methods and approaches to perform a sales forecast. Part of the data is used to train a neural network algorithm, with backpropagation for some methods, step by step, where shallow nets face selected scenarios, with different space-time data considerations.

In each method, by using a sum of square differences, and a peak search procedure, a reasonable quality in the obtained abstract representations is pursued. First, an auto-encoder is trained to develop in its hidden layer neural data abstractions about a random-moving window. Thereafter by using the abstraction of the net plus recently captured information, a second shallow net is trained to produce its own one-day ahead estimates, using new timing and data procedures. After training, the whole stacked system's performance is compared with the naive forecast scenario's mean square error and if it's a better value, the method is used to produce stable daily forecasting for assorted products and periods. The system has been tested in real-time with real data.

Contents

List of Figures

List of Tables

List of Abbreviations

ARIMA	Auto Regression Integrated Moving Average
LSTM	Long Short Term Memory
MSE	Mean Square Error
KPI	Key Performance Indicator
API	Application Programming Interface

Chapter 1

Introduction

1.1 Background

One of the responsibilities of pharmacies in Uganda is to have a minimum stock of medicines.This ensures patients can have it when prescribed.

In addition, pharmacies need to get a good forecast of the medication needs due to the short term validity of many medicines and the need to control stock levels.This avoids excessive costs and loss of customers due to stock outages.

A good sales forecast is usually associated with striking a balance between stock costs and adequate satisfaction of customer demand.People act on the basis of forecasting models whether they are on paper or in their heads. You are better off quantifying these estimations so you can discuss them rationally as opposed to making them based on intuition.

To specific case of pharmacies in Uganda, the problem is of particular importance due to the short cycle life of many products and the importance of quality which is in turn strongly linked to public health.

During our research study of Soteria Pharmacy procurement process with an interview of Ms.Brenda the incharge of this, we realised she uses personal judgement of current stock levels athand and the rate at which people come in to ask for a ceratin drug then determine how much more stock should be purchased.Given her difficulties in accurately predicting the future sales,this report explores the product sales forecast at the indiviual level of this Pharmacy.I made a forecast for 5 sold drugs.The forecast was based on analysis of historical data for a period of 24 months and future results analyses of 50 days determined.ARIMA and

LSTM methods were used in the sales forecast of which the best model was recommended including a conclusive way of improving our results.

1.2 Aims and Objectives

The main aim of the research project was to precisely predict sales of drugs and medical supplies.

The specific objectives were:

- To collect datasets of previous sales and purchases from Soteria Pharmacy.
- To train and validate datasets with ARIMA and LSTM models.
- To optimize the best model algorithm for accurate performance.
- To develop a web API for pharmacies.

1.3 Contributions of the Project

The following major contributions have been accomplished during the course of this research:

1. Performed a data cleaning task, analysis and training with the various Machine learning models and came up with a graphical output.
2. Developed a robust neural network model that could accurately predict future sales for a period of 50 days using a 2 year period worth of sales and purchases information.
3. Proposed a multivariate input approach with other characteristics such as precipitational weather information and promotional sales.

1.4 Organization of the Report

This report has 3 chapters.

- Introduction
- Analysis and Findings which characterizes essential concepts, an overview of the categories and description of the main methods associated with the several techniques of time series analysis.

- Conclusion presents some final considerations and future work proposals.

Chapter 2

Literature Review

2.1 Introduction

People act on the basis of forecasting models whether they are on paper or in their heads. One is better off quantifying these estimations so as to discuss them rationally as opposed to making them based on intuition. For pharmaceutical distribution companies, it is essential to get good estimates of drugs, due to the short shelf life of many medicines and the need to control stock levels, to avoid excessive inventory costs while guaranteeing customer demand satisfaction, and thus decreasing the possibility of a loss of customers due to stock outages. Stock management, transportations, and financial spends contain a high percentage of total pharmaceutical companies' expenses. As a rule, companies pay immediately when buy medications from manufacturers, and then sales compensate these spend gradually. This gap is a danger of unplanned expenses to occur. Therefore, the majority of distributors in this industry look for modern and precise forecasting methods of future sales to decrease purchase and storage costs and to increase profit by meeting clients' needs timely. Common existing forecasting methods are ineffective for pharmaceutical companies because these methods require a large dataset of each medication sales. In its turn, medications are constantly replaced by analogs or refreshed to enhance pharmacological effects or reduce collateral effects.

2.2 Literature Review

Consequently, it is required to build an accurate forecasting model of pharmaceutical preparations sales using one of the machine learning methods taking into account constant medications refreshment and a lack of previous sales data [1].

Before digital technology dominated the world, the forecasting process was done manually by experienced individuals in the domain. This intuition required a lot of experience and was prone to error. Due to this reason, they started realizing the need for automating the pharmacy sales forecasting process. Thus, research and experiments were carried out with statistical, machine learning, deep learning, and ensemble techniques to achieve more accurate sales forecasts. [2] Algorithms to illustrate inherent laws in large amounts of data and to forecast future data patterns have been researched since 1920. However no breakthroughs till 1980. [3] In the deep learning world, state-of-the-art performance has gained a good reputation in fields like object recognition, [4] speech recognition [5], natural language processing [6], physiological effect modeling [7], and many others. More recently papers on time-series prediction or classification with deep neural networks have been reported. [8–11] Aimed at obtaining inherent laws of historical data series in pharmaceutical sales, and forecasting the demand, controlling the inventory, reducing the costs, and improving the service level, this paper designs research on the data records from a pharmacy and presents different forecasting algorithms. The testing results provide support for the fact that these algorithms greatly improve the forecast accuracy from the naive forecasting method.

2.3 Summary

Chapter 3

Methodology

3.1 Machine Learning

3.1.1 Introduction

Machine learning is a subfield of artificial intelligence (AI). The goal of machine learning generally is to understand the structure of data and fit that data into models that can be understood and utilized by people.

Although machine learning is a field within computer science, it differs from traditional computational approaches. In traditional computing, algorithms are sets of explicitly programmed instructions used by computers to calculate or problem solve. Machine learning algorithms instead allow for computers to train on data inputs and use statistical analysis in order to output values that fall within a specific range. Because of this, machine learning facilitates computers in building models from sample data in order to automate decision-making processes based on data inputs.

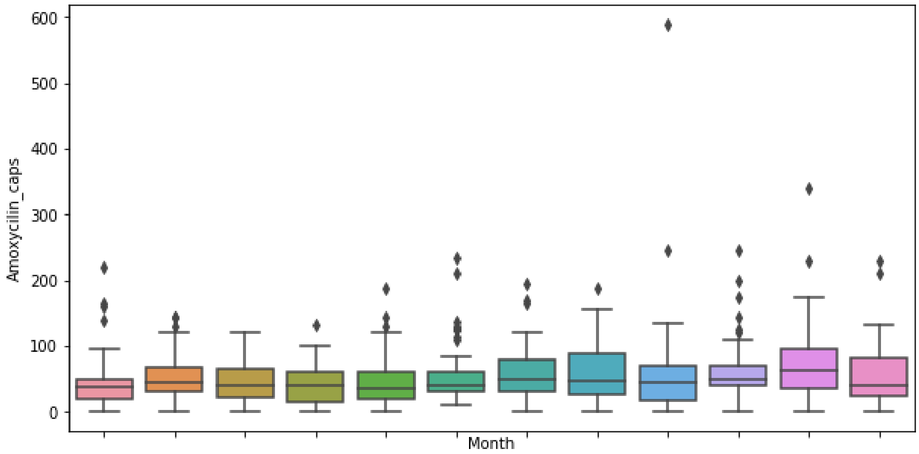
Any technology user today has benefitted from machine learning. Facial recognition technology allows social media platforms to help users tag and share photos of friends. Optical character recognition (OCR) technology converts images of text into movable type. Recommendation engines, powered by machine learning, suggest what movies or television shows to watch next based on user preferences. Self-driving cars that rely on machine learning to navigate may soon be available to consumers.

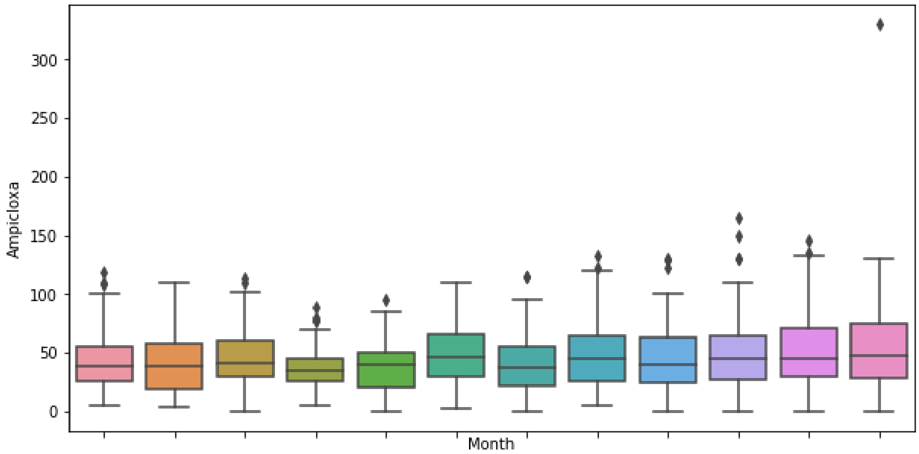
Machine learning is a continuously developing field. Because of this, there are some considerations to keep in mind as you work with machine learning methodologies, or analyze the impact of machine learning processes. [12]

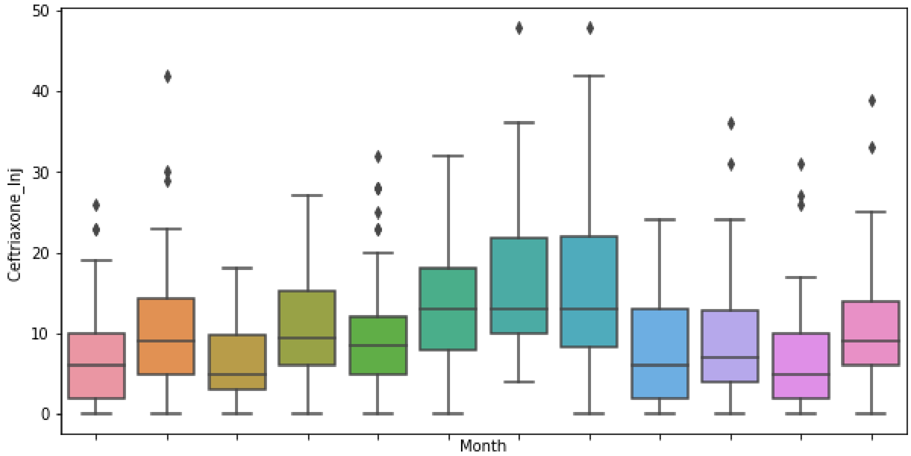
3.2 Materials and Methods

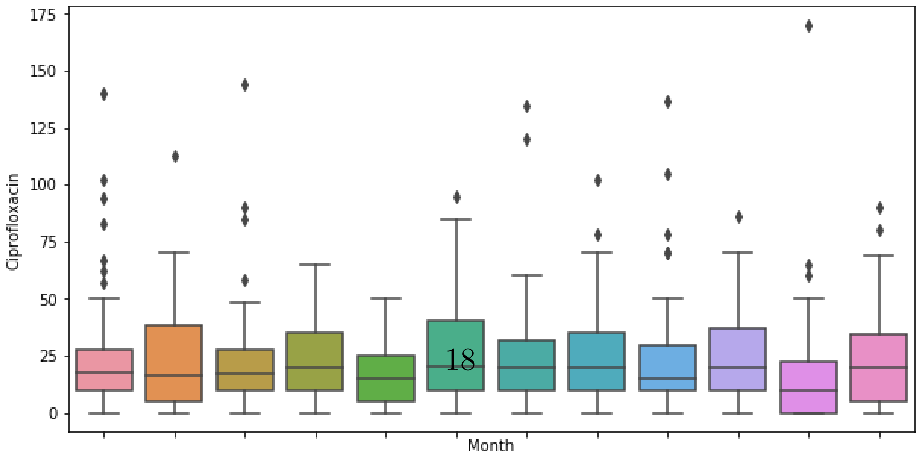
We analyzed daily sales data for a period of 2 years by arranging it in a proper format to easily perform predictions with the same data file using Microsoft excel. It was imported into the google colab jupyter notebook by a clone from GitHub, then using pandas passed into a variable for further processing. From box-plots, Cotrimoxazole had more outliers than the rest of the drugs which made it harder to predict future sales.

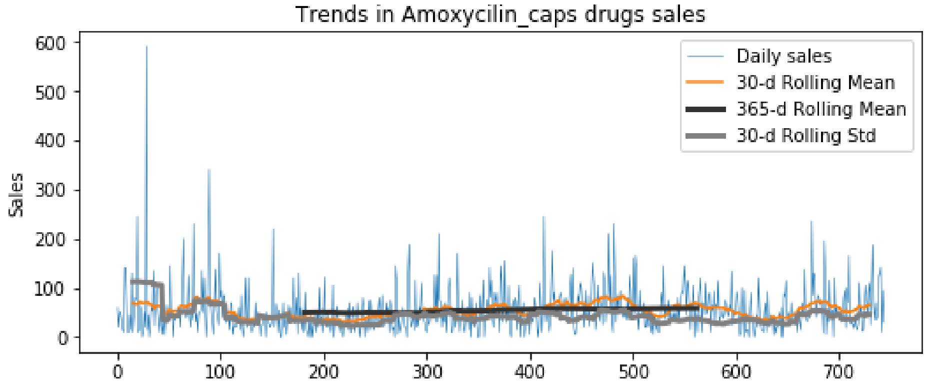
We also checked for presence of seasonality and trend with a 30 day and 365 day rolling mean graph.

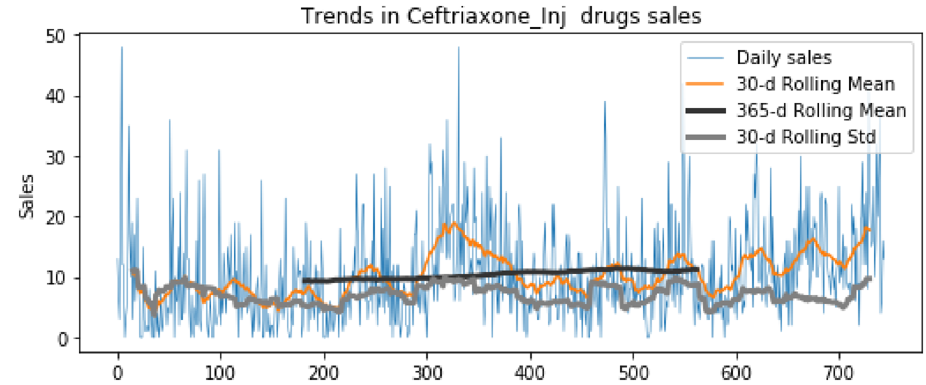

Amoxycilin_caps
0
100
200
300
400
500
600
Month

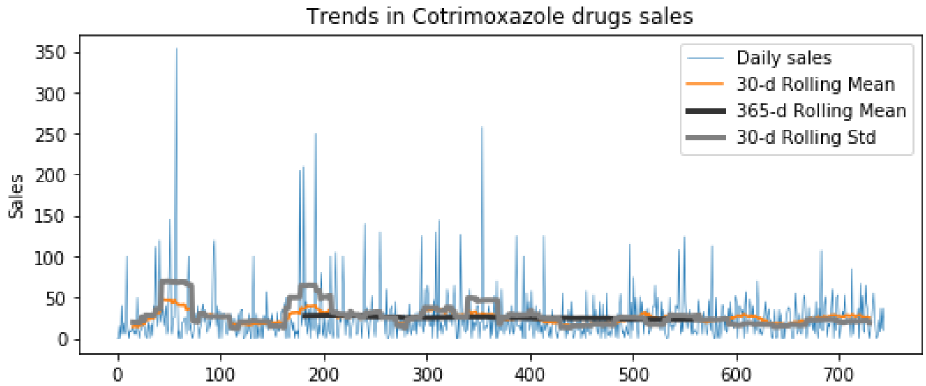

Ampicloxa
0
50
100
150
200
250
300
Month

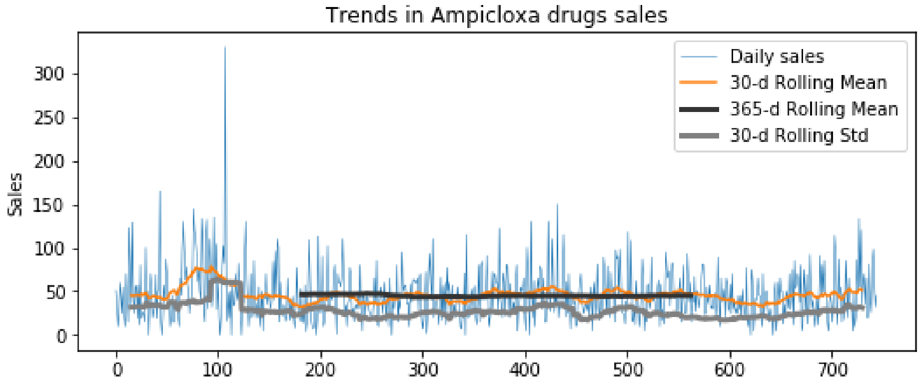

Ceftriaxone_Inj
0
10
20
30
40
50
Month

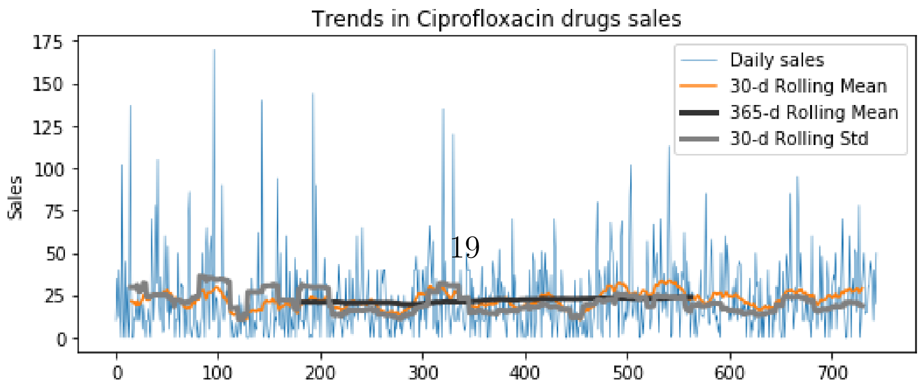

Ciprofloxacin
0
25
50
75
100
125
150
175
Month


Trends in Amoxycilin_caps drugs sales
Sales
Daily sales
30-d Rolling Mean
365-d Rolling Mean
30-d Rolling Std


Trends in Ceftriaxone_Inj drugs sales
Sales
Daily sales
30-d Rolling Mean
365-d Rolling Mean
30-d Rolling Std


Trends in Cotrimoxazole drugs sales
Sales
Daily sales
30-d Rolling Mean
365-d Rolling Mean
30-d Rolling Std


Trends in Ampicloxa drugs sales
Sales
Daily sales
30-d Rolling Mean
365-d Rolling Mean
30-d Rolling Std


Trends in Ciprofloxacin drugs sales
Sales
Daily sales
30-d Rolling Mean
365-d Rolling Mean
30-d Rolling Std

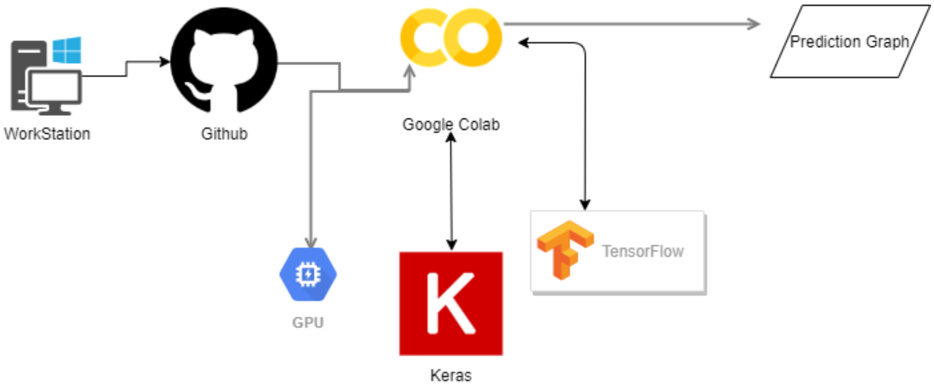


Figure 3.3: Summary workflow chart.

Included Components • Keras: The Python Deep Learning library. • Tensorflow: An open-source software library for Machine Intelligence. • Github:An open-source platform to store code. • GPU:A processing unit for computationally intense algorithms.

3.3 Time Series Forecast

A time series is usually modelled through a stochastic process Y(t), i.e. a sequence of random variables. In a forecasting setting we find ourselves at time t and we are interested in estimating Y(t+h), using only information available at time t. [13] Anything that is observed sequentially over time is a time series. When forecasting time series data, the aim is to estimate how the sequence of observations will continue into the future. Forecasting involves taking models fit on historical data and using them to predict future observations.

Descriptive models can borrow for the future (i.e. to smooth or remove noise), they only seek to best describe the data. The skill of a time series forecasting model is determined by its performance at predicting the future. This is often at the expense of being able to explain why a specific prediction was made, confidence intervals and even better understanding the underlying causes behind the problem.

Components of Time Series

Time series analysis provides a body of techniques to better understand a dataset.

Perhaps the most useful of these is the decomposition of a time series into 4 constituent parts:

1. Level. The baseline value for the series if it were a straight line.

2. Trend. The optional and often linear increasing or decreasing behavior of the series

over time.

3. Seasonality. The optional repeating patterns or cycles of behavior over time.

4. Noise. The optional variability in the observations that cannot be explained by the model. All time series have a level, most have noise, and the trend and seasonality are optional.

3.4 Data and Results

3.4.1 Naïve forecasting method

This was the baseline method since it is what most pharmacists use during the normal day to day operation of their business for drug replenishments. Below are the results in graphical format.Our forecast values were based on the last 50 days of our 2 year period data sample. This uses a day time shift of the previous sales to predict future sales. As a case study for soteria pharmacy in luweero district, they normally use intuition similar to this whereby the next day purchases are made basic on the previous day sales and hoping the same trend. With our time day shift results , we modelled a naive forecasting scenario and were able to assume this as the result of what is currently being used considering most pharmacy procurement personnel use intuition.Figure 3.4 shows the plots obtained from the naive forecast scenario.

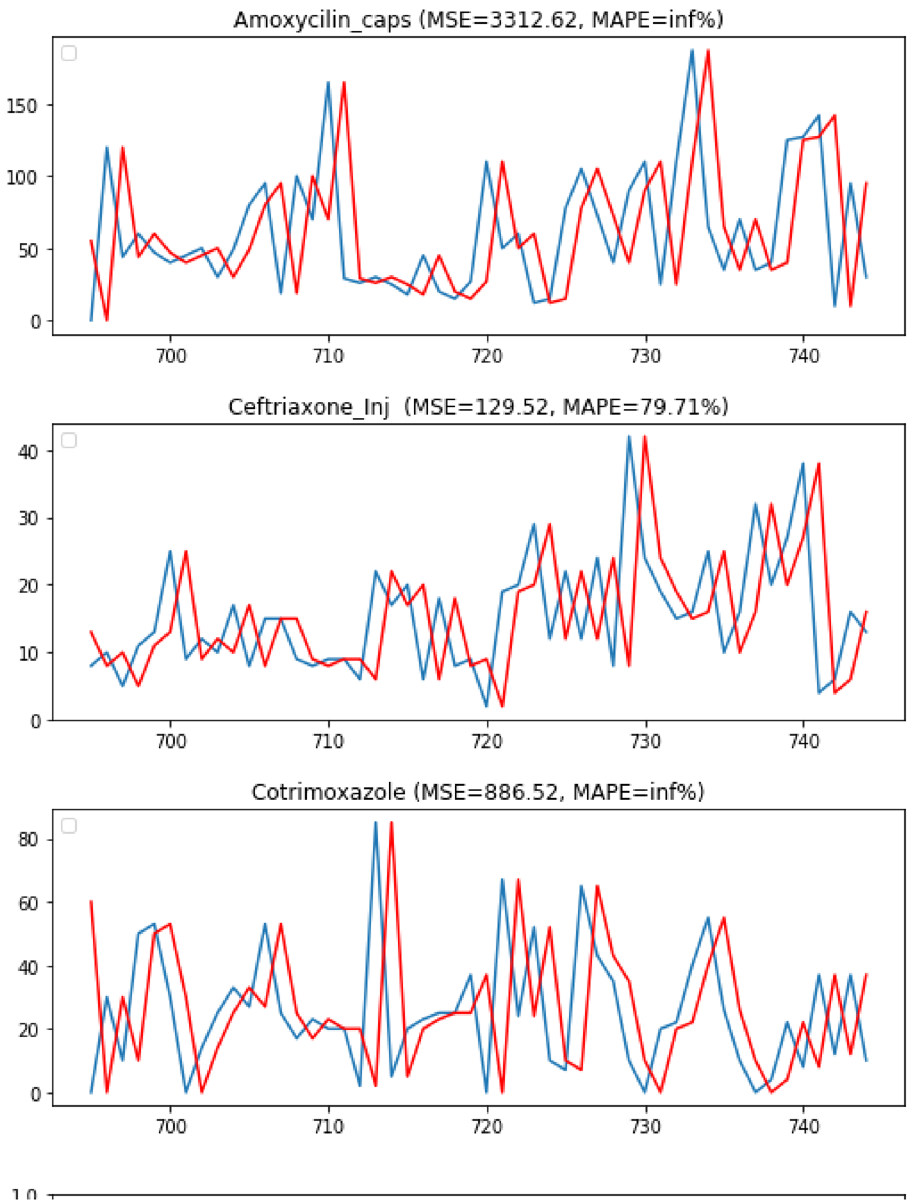


Figure 3.4: Illustration plot of the naïve forecasting results.

3.4.2 ARIMA method

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 the lagged forecast errors, so that equation can be used to forecast future values.

An ARIMA model is characterized by 3 terms: p, d, q

where,

p is the order of the Auto Regression(AR) term

q is the order of the Moving Average(MA) term

d is the number of differencing required to make the time series stationary

A pure Auto Regressive (AR only) model is one where Yt depends only on its own lags. That is, Yt is a function of the 'lags of Yt'.

Likewise a pure Moving Average (MA only) model is one where Yt depends only on the lagged forecast errors.

First, method arma_order_select_ic was used to determine initial p and q parameters. The method computes Akaike's Information Criterion (AIC) for many ARIMA models and chooses the best configuration. However, AIC is not used to score accuracy of the forecasting methods in this research. Mean squared error is used instead. For that, reason, grid search optimization method was applied, where different combinations of the hyper-parameters were used to calculate MSE and then, the combination producing the least MSE was chosen as optimal. Grid search optimization for rolling forecast produced the following best combinations of the hyper-parameters:

Amoxyclin_caps – Best ARIMA(1,0,0) MSE=1524.745

3.4.3 Forecasting with LSTM methods

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network capable of learning order dependence in sequence prediction problems.
Recurrent neural networks are different from traditional feed-forward neural networks.

This difference in the addition of complexity comes with the promise of new behaviors that the traditional methods cannot achieve.
Long-term forecasting validation has been done with three LSTM configurations: Vanilla LSTM, Stacked LSTM and Bi-directional LSTM. Relu activation function was used, optimizer was Adam and loss function was Mean Squared Error. The best results were achieved with training the model in 400 epochs. Before fitting, all data was standardized (rescaled in interval -1,1) and transformed to data for supervised problem. Number of past observations

tested in input sequences was 5.

Vanilla LSTM

This is where we have one hidden layer within the neural network. In order to get reproducible results in forecasting with LSTM, following values are fixed: seed value, 'PYTHON-HASHSEED' environment variable, Python's, numpy's and Tensorflow's built-in pseudo-random generators. A new global Tensorflow session is configured.Below are the results from the vanilla LSTM model. This model is unidirectional in the sense that the current output is only influenced by the past states and the current input.

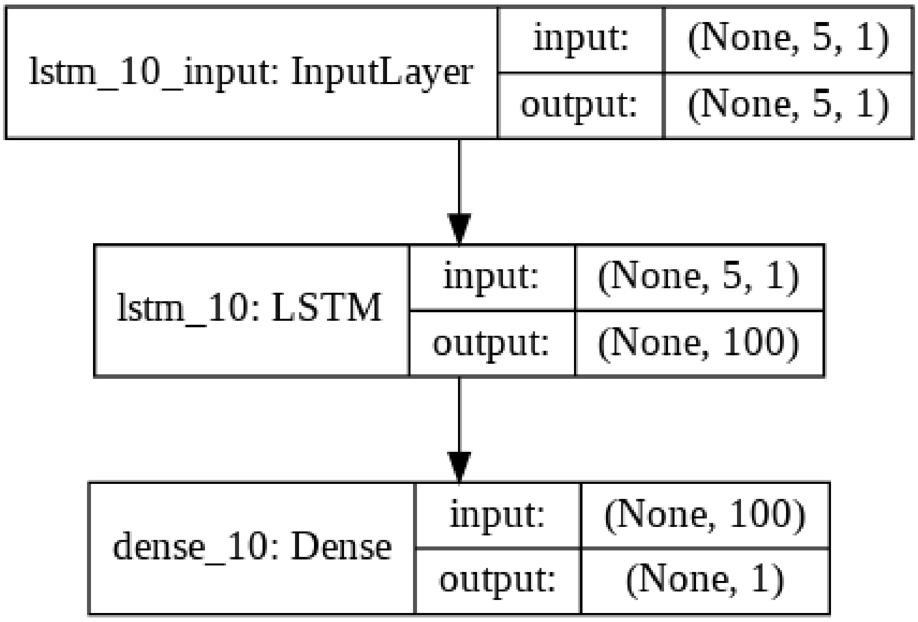


Figure 3.5: Vanilla LSTM model Graphical Layout.

```
Model: "sequential_10"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_10 (LSTM)               (None, 100)               40800
_________________________________________________________________
dense_10 (Dense)             (None, 1)                 101
=================================================================
Total params: 40,901
Trainable params: 40,901
Non-trainable params: 0
_________________________________________________________________
None
```

Figure 3.6: Vanilla LSTM model Summary.

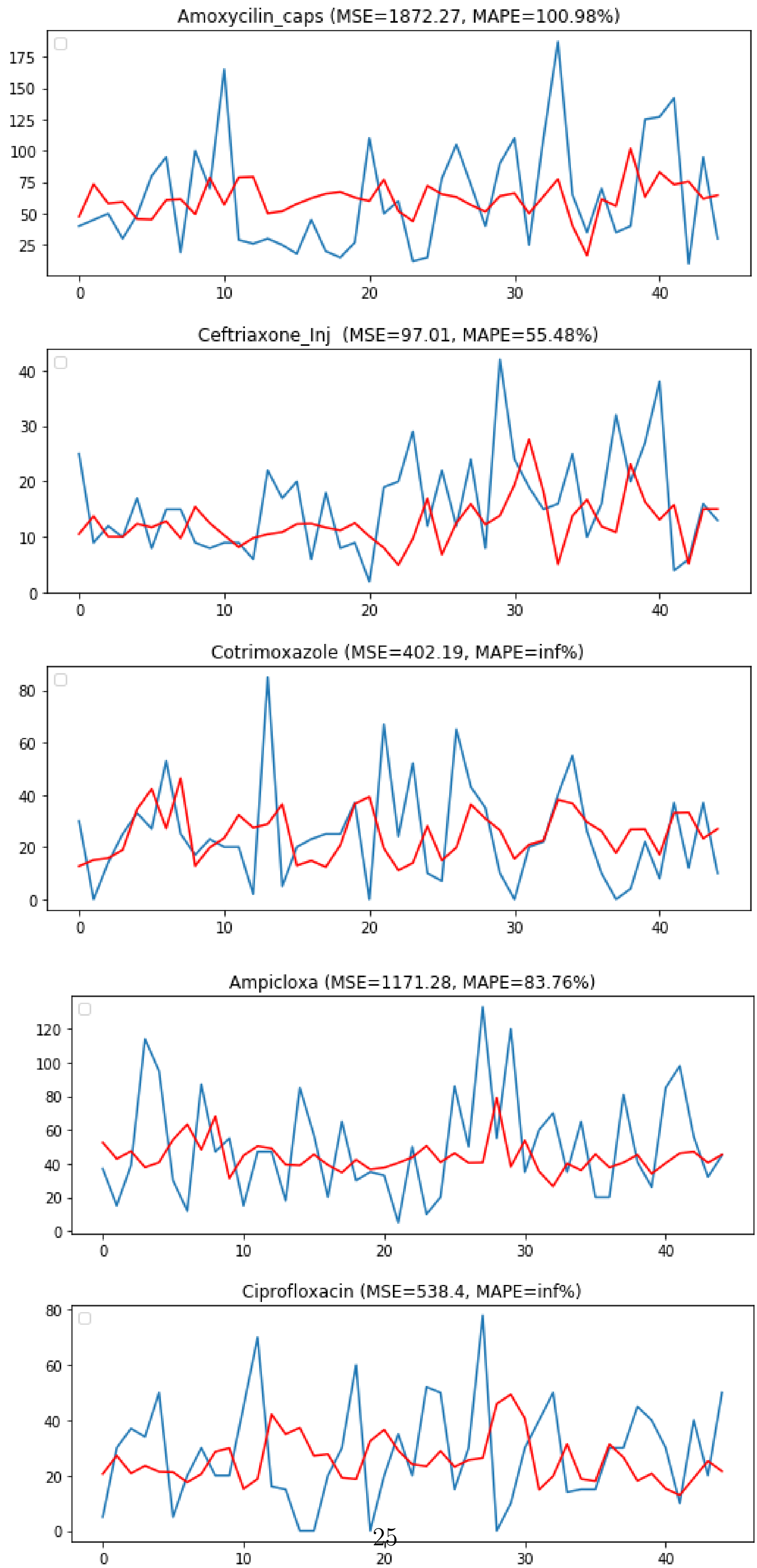


Figure 3.7: Vanilla LSTM model results.

Stacked LSTM method

This is also a unidirectional model where we have more than one hidden layer within the neural network.

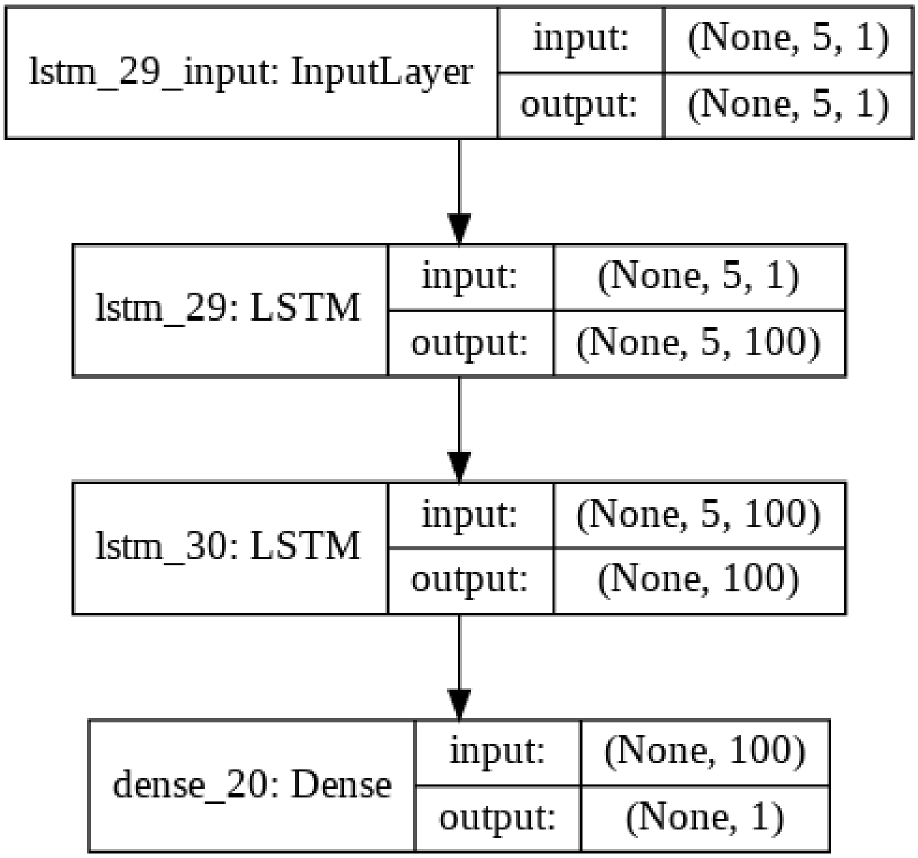
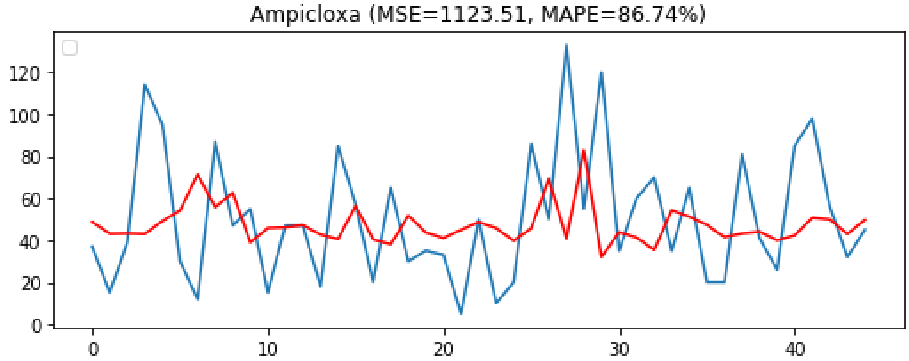
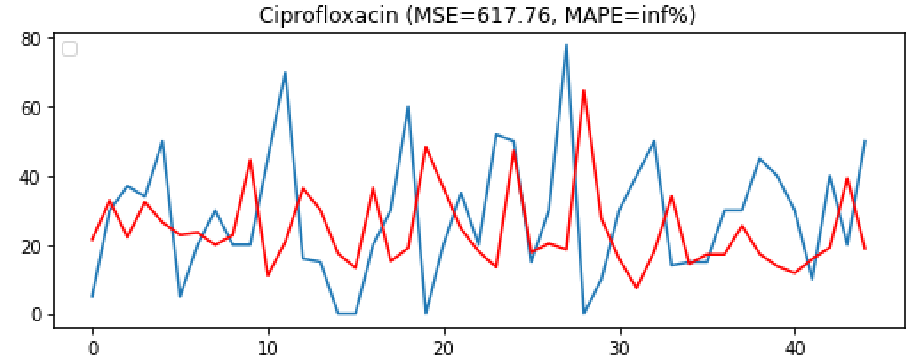


Figure 3.8: Stacked LSTM model Graphical Layout.

```
Model: "sequential_20"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_29 (LSTM)               (None, 5, 100)            40800
_________________________________________________________________
lstm_30 (LSTM)               (None, 100)               80400
_________________________________________________________________
dense_20 (Dense)             (None, 1)                 101
=================================================================
Total params: 121,301
Trainable params: 121,301
Non-trainable params: 0
_________________________________________________________________
None
```

Figure 3.9: Stacked LSTM model Summary.


Ampicloxa (MSE=1123.51, MAPE=86.74%)


Ciprofloxacin (MSE=617.76, MAPE=inf%)

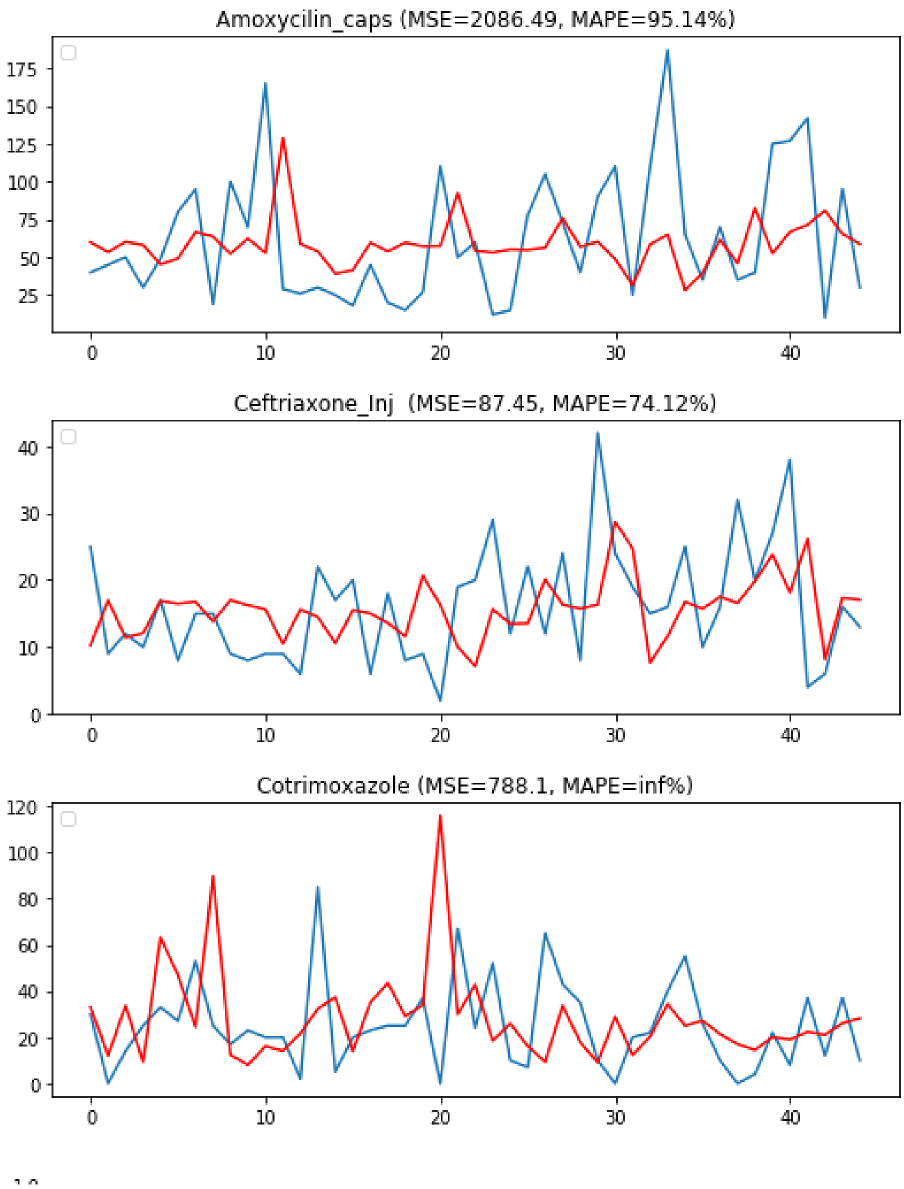


Figure 3.10: Stacked LSTM model results.

Bidirectional LSTM

In bidirectional RNNs, future states can also influence the present state and the past states by allowing information to flow backward. Past outputs are updated as needed depending on the new information received.

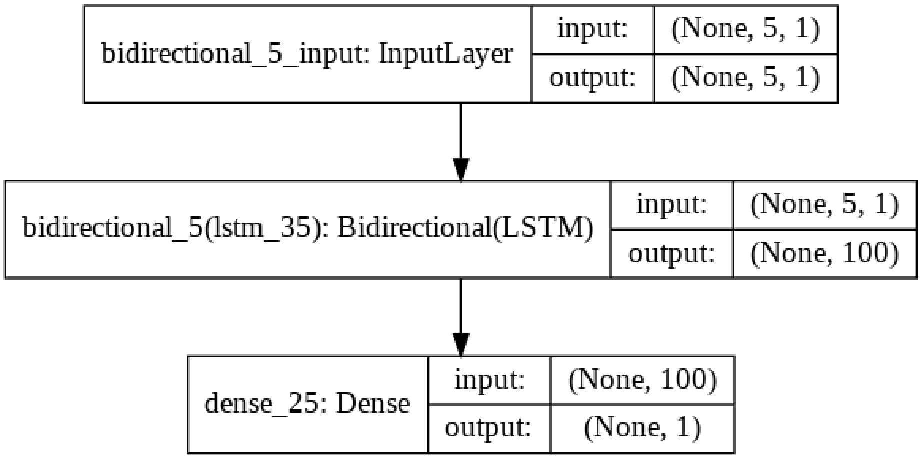


Figure 3.11: Bidirectional LSTM model Graphical Layout.

```
Model: "sequential_25"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
bidirectional_5 (Bidirection (None, 100)               20800
_________________________________________________________________
dense_25 (Dense)             (None, 1)                 101
=================================================================
Total params: 20,901
Trainable params: 20,901
Non-trainable params: 0
_________________________________________________________________
None
```

Figure 3.12: Bidirectional LSTM model Summary.

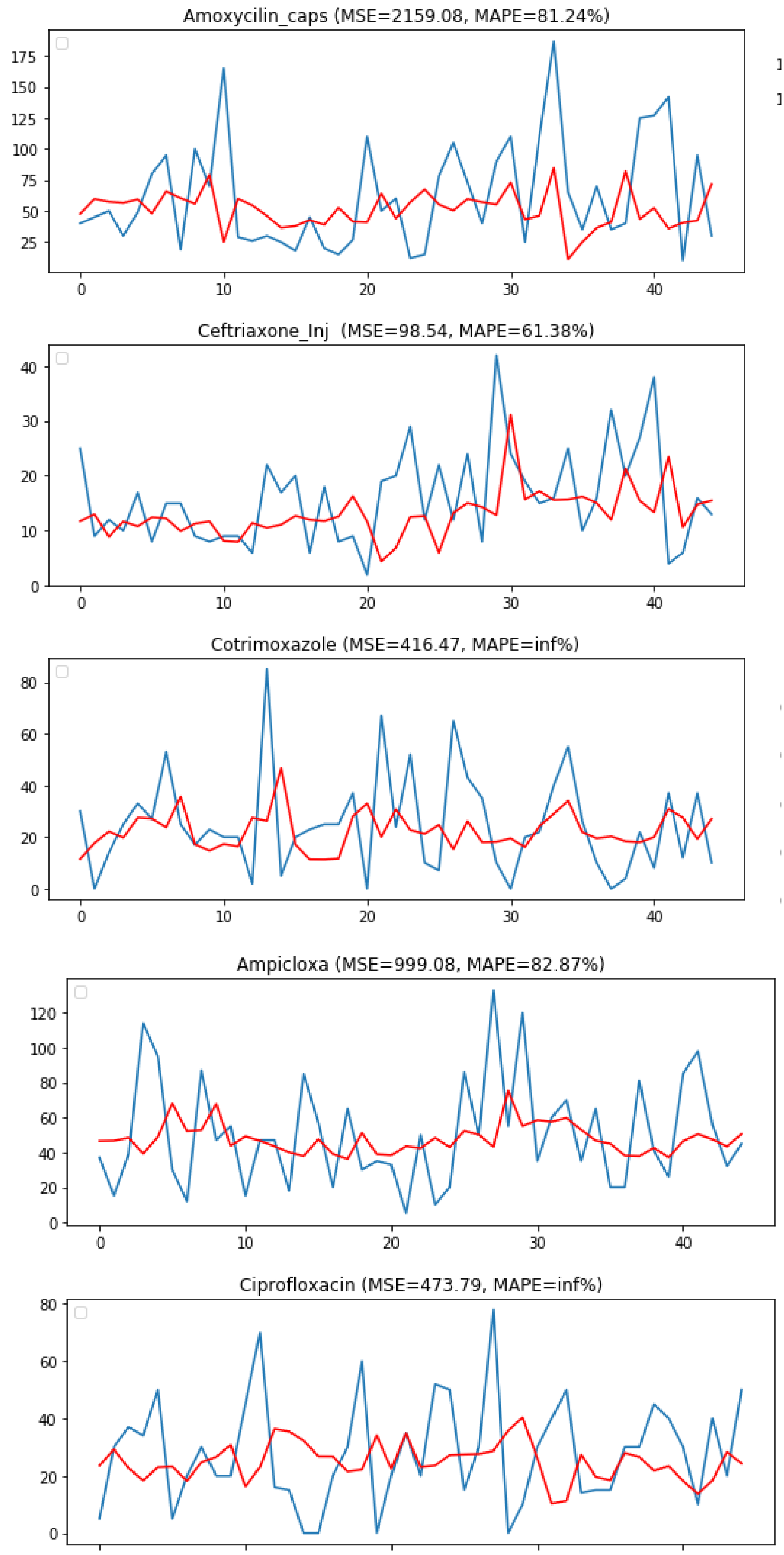


Figure 3.13: Bidirectional LSTM model results.

3.5 Web API

I then built a web API that can be used by pharmacy managers using the streamlit python framework. One is able to select one of the various machine learning models in a drop down selection menu as shown below.

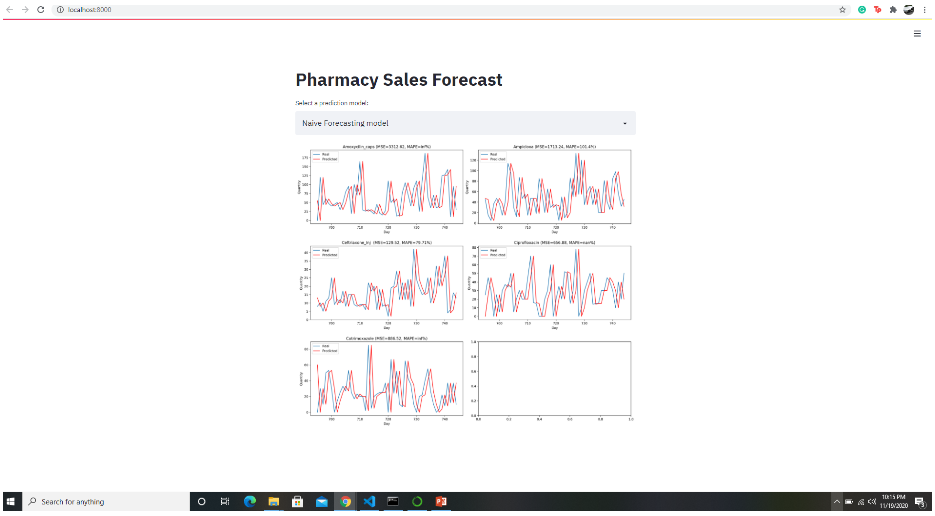


Figure 3.14: Web App API.

Chapter 4

Discussion

Figure 4.1 shows that all the methods used have a less MSE than the currently used Naive forecasting methods.

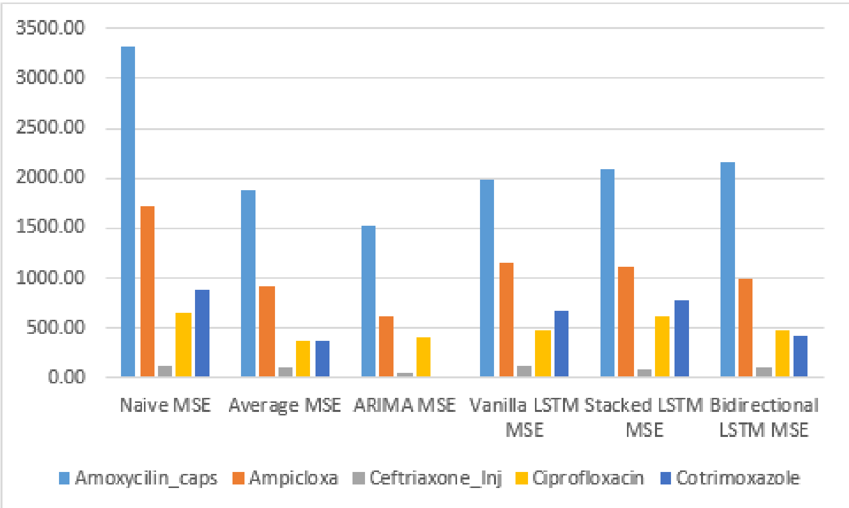


Figure 4.1: MSE overall results.

However with more data of about 5 years or more, better results can be obtained for the LSTM methods .

Chapter 5

Conclusions and Future Works

5.1 Conclusions

5.1.1 Summary of findings

This shows that the ARIMA gives the lowest MSE compared to the rest of the models.

5.1.2 Limitations of Study

Among the shortcomings of our research are the reluctance of most pharmacies especially those able to provide us with the long time-span data to avail us with this information considering it is company sensitive and only known to a specific set of employees.

5.1.3 Recommendations

We suggest that more data sets are used with a longer lifetime span of about 5 years worth data.We also suggest the incorporation of other secondary dependent features that affect the sales such as weather patterns, promotional sales, marketing elements in order to get more accurate results.

5.1.4 Acknowledgements

I thank Dr.Andrew Katumba for the support offered at every stage of this project from day 1, he is a key component to enabling us see this idea to fruitation. We also acknowledge the management of Soteria pharmacy in Luweero district. Specifically Ms. Brenda that availed us with the required information and data plus explaining the entire workflow of the pharmacy purchasing process to us.

5.2 Ideas for Future Work

Incorparating a multivariate model that makes use of various input data to forecast the results such as weather data ,promotion sales data e.t.c

Bibliography

[1] S. H. Perera, N. Jihan, M. Jayasinghe and Srinath "Sales forecasting using multivariate long short term memory network models," *PeerJ PrePrints,* vol. 7, pp. e27712, 2019.

[2] Zadeh Khalil, N. Sepehri, M. M., Farvaresh"Intelligent sales prediction for pharmaceutical distribution companies, A data mining based approach," *Mathematical Problems in Engineering,* 2014.

[3] Lian-hong Cai, "Rough Set in Neural Network," *Computer Engineering,* Vol 27,No.5 pp65-69,2001.

[4] Alex Krizhevsky, I Sutskever, and GE Hinton, "Imagenet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems," *object recognition,* pages 1–9, 2012

[5] O Abdel- Hamid and A Mohamed. "Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition," *Acoustics, Speech, and Signal Processing,* 2012

[6] R. Collobert and Jason Weston. "A unified architecture for natural language processing: Deep neural networks with multitask learning," *Proceedings of the 25th international conference on Machine learning,* 2008

[7] HP Martinez. "Learning deep physiological models of affect." *Computational Intelligence Magazine,* (April):20–33, 2013. (physiological affect modelling)

[8] Atiya, Amir F. , Gayar, Neamat El and El- Shishiny, Hisham(2010) "An Empirical Comparison of Machine Learning Models for Time Series Forecasting." *Econometric Reviews,* 29: 5, 594-621

[9] S. Prasad, P. Prasad. "Deep Recurrent Neural Networks for Time Series Prediction"

[10] "DeepLearningforEvent-DrivenStockPrediction Mladen Dalto 'Deep neural networks for time series prediction with applications in ultra-short-term wind forecasting" *2015 IEEE International Conference on Industrial Technology (ICIT)*

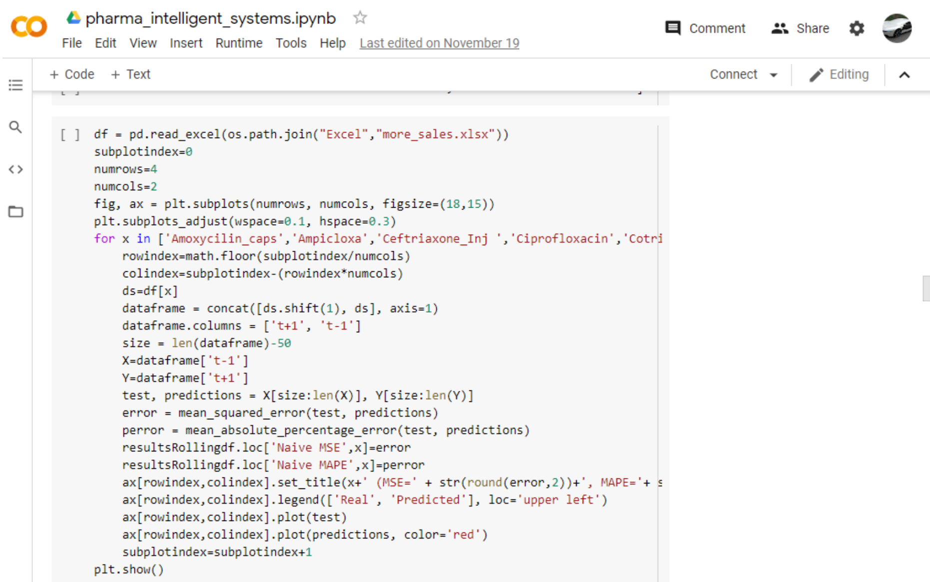
[11] XiaoDing, YueZhang, TingLiu, JunwenDuan, Mladen Dalto, "Deep neural networks for time series prediction with applications in ultrashort-term wind forecasting time series I" *University of Zagreb, Faculty of Electrical Engineering and Computing*

[12] Lisa Tagliaferri"An Introduction to Machine Learning" *https://www.digitalocean.com/community/tutorials/an-introduction-to-machine-learning*

[13] Davide Burba"An overview of time series forecasting models" *https://towardsdatascience.com/an-overview-of-time-series-forecasting-models-a2fa7a358fcb*

Appendix A

Appendix




```
[ ] df = pd.read_excel(os.path.join("Excel","more_sales.xlsx"))
    subplotindex=0
    numrows=4
    numcols=2
    fig, ax = plt.subplots(numrows, numcols, figsize=(18,15))
    plt.subplots_adjust(wspace=0.1, hspace=0.3)
    for x in ['Amoxycilin_caps','Ampicloxa','Ceftriaxone_Inj ','Ciprofloxacin','Cotri
        rowindex=math.floor(subplotindex/numcols)
        colindex=subplotindex-(rowindex*numcols)
        ds=df[x]
        dataframe = concat([ds.shift(1), ds], axis=1)
        dataframe.columns = ['t+1', 't-1']
        size = len(dataframe)-50
        X=dataframe['t-1']
        Y=dataframe['t+1']
        test, predictions = X[size:len(X)], Y[size:len(Y)]
        error = mean_squared_error(test, predictions)
        perror = mean_absolute_percentage_error(test, predictions)
        resultsRollingdf.loc['Naive MSE',x]=error
        resultsRollingdf.loc['Naive MAPE',x]=perror
        ax[rowindex,colindex].set_title(x+' (MSE=' + str(round(error,2))+', MAPE='+ s
        ax[rowindex,colindex].legend(['Real', 'Predicted'], loc='upper left')
        ax[rowindex,colindex].plot(test)
        ax[rowindex,colindex].plot(predictions, color='red')
        subplotindex=subplotindex+1
    plt.show()
```

Figure A.1: My Naive forecast code Base on Google Colab

```
[ ]  r=['Amoxycilin_caps']
     for x in r:
         rowindex=math.floor(subplotindex/numcols)
         colindex=subplotindex-(rowindex*numcols)
         X=df[x].values
         scaler = MinMaxScaler(feature_range = (0, 1))
         X=scaler.fit_transform(X.reshape(-1, 1))
         X_train,y_train=split_sequence(X[0:size], n_steps)
         X_test,y_test=split_sequence(X[size:len(df)], n_steps)
         X_train = X_train.reshape((X_train.shape[0], X_train.shape[1], n_features))

         model = Sequential()
         model.add(LSTM(1000, activation='relu', return_sequences=True, input_shape=(r
         model.add(LSTM(1000, activation='relu'))
         model.add(Dense(1))
         model.compile(optimizer='adam', loss='mse')
         history = model.fit(X_train, y_train, epochs=100, verbose=1, validation_data=
         train[str(sd)] = history.history['loss']
         val[str(sd)] = history.history['val_loss']
         az[rowindex,colindex].set_title(x)
         az[rowindex,colindex].plot(train[str(sd)])
         az[rowindex,colindex].plot(val[str(sd)], color='red')
         sd=sd+1

         X_test = X_test.reshape((len(X_test), n_steps, n_features))
         predictions = model.predict(X_test, verbose=0)
         y_test=scaler.inverse_transform(y_test)
         predictions = scaler.inverse_transform(predictions)
         error = mean_squared_error(y_test, predictions)
```

Figure A.2: My Vanilla LSTM forecast code Base on Google Colab

```
     warnings.filterwarnings('ignore')
[ ]
     #r=['Amoxycilin_caps','Ampicloxa','Ceftriaxone_Inj ','Ciprofloxacin','Cotrimoxazole']
     r=['Amoxycilin_caps']
     for x in r:
         rowindex=math.floor(subplotindex/numcols)
         colindex=subplotindex-(rowindex*numcols)
         X=df[x].values
         scaler = MinMaxScaler(feature_range = (0, 1))
         X=scaler.fit_transform(X.reshape(-1, 1))
         X_train,y_train=split_sequence(X[0:size], n_steps)
         X_test,y_test=split_sequence(X[size:len(df)], n_steps)
         X_train = X_train.reshape((X_train.shape[0], X_train.shape[1], n_features))

         model = Sequential()
         model.add(Bidirectional(LSTM(1000, activation='relu'), input_shape=(n_steps, n_features)))
         model.add(Dense(1))
         model.compile(optimizer='adam', loss='mse', metrics=['accuracy'])
         cb.logs=[]
         history = model.fit(X_train, y_train, epochs=100, verbose=1, callbacks=[cb], validation_data=(X_test, y_te
         train[str(sd)] = history.history['loss']
         val[str(sd)] = history.history['val_loss']
         az[rowindex,colindex].set_title(x)
         az[rowindex,colindex].plot(train[str(sd)])
         az[rowindex,colindex].plot(val[str(sd)], color='red')
         sd=sd+1
         #loss, acc = model.evaluate(X_test, y_test)
         #print(sum(cb.logs))
         #print(acc*100.0)
         X_test = X_test.reshape((len(X_test), n_steps, n_features))
         predictions = model.predict(X_test, verbose=0)
         y_test=scaler.inverse_transform(y_test)
         predictions = scaler.inverse_transform(predictions)
         error = mean_squared_error(y_test, predictions)
         perror = mean_absolute_percentage_error(y_test, predictions)
         resultsLongtermdf.loc['Bidirectional LSTM MSE',x]=error
         resultsLongtermdf.loc['Bidirectional LSTM MAPE',x]=perror
         ax[rowindex,colindex].set_title(x+' (MSE=' + str(round(error,2))+', MAPE='+ str(round(perror,2)) +'%)')
```

Figure A.3: My Stacked LSTM forecast code Base on Google Colab

	Amoxycilin_caps	Ampicloxa	Ceftriaxone_Inj	Ciprofloxacin	Cotrimoxazole
Average MSE	1882.194144	927.985467	102.020312	363.422187	373.897875
Average MAPE	inf	94.288002	48.285348	inf	inf
ARIMA MSE	0.000000	0.000000	0.000000	0.000000	0.000000
ARIMA MAPE	0.000000	0.000000	0.000000	0.000000	0.000000
AutoARIMA MSE	0.000000	0.000000	0.000000	0.000000	0.000000
AutoARIMA MAPE	0.000000	0.000000	0.000000	0.000000	0.000000
Prophet MSE	0.000000	0.000000	0.000000	0.000000	0.000000
Prophet MAPE	0.000000	0.000000	0.000000	0.000000	0.000000
Vanilla LSTM MSE	1992.988497	1152.440552	114.375701	477.080848	678.311004
Vanilla LSTM MAPE	89.127919	87.236741	69.777434	inf	inf
Stacked LSTM MSE	2086.485384	1123.511486	87.453323	617.756463	788.095327
Stacked LSTM MAPE	95.139783	86.743554	74.122829	inf	inf
Bidirectional LSTM MSE	2159.075832	999.079763	98.544152	473.785420	416.467764
Bidirectional LSTM MAPE	81.239085	82.872384	61.377742	inf	inf

Figure A.4: Results of the machine learning forecast