National College of Ireland

Empirical Evaluation of various Deep Neural Network Architectures for Time Series Forecasting

MSc Research Project
Data Analytics

Chetan Sharma
x16131819

School of Computing
National College of Ireland

Supervisor: Michael Bradford

National College of Ireland
Project Submission Sheet – 2017/2018
School of Computing

Student Name:	Chetan Sharma
Student ID:	X16131819
Programme:	Msc Data Analytics
Year:	2017
Module:	Research Project
Lecturer:	Michael Bradford
Submission Due Date:	11/12/2017
Project Title:	Empirical Evaluation of Various Deep Neural Network Architectures for Time Series Forecasting
Word Count:	6636

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are encouraged to use the Harvard Referencing Standard supplied by the Library. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action. Students may be required to undergo a viva (oral examination) if there is suspicion about the validity of their submitted work.

Signature:	
Date:	11th December 2017

PLEASE READ THE FOLLOWING INSTRUCTIONS:

1. Please attach a completed copy of this sheet to each project (including multiple copies).
2. **You must ensure that you retain a HARD COPY of ALL projects**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. Please do not bind projects or place in covers unless specifically requested.
3. Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Empirical Evaluation of various Deep Neural Network Architectures for Time Series Forecasting

Chetan Sharma
x16131819
MSc Research Project in Data Analytics

11th December 2017

Abstract

Time series forecasting is regarded amongst the top 10 challenges in data mining. Lately, deep learning based models have garnered a lot of attention from researchers in time series forecasting. However, which deep neural network architecture is most appropriate in time series forecasting domain has not been researched extensively.In this research performance of 4 deep neural network architectures MLP (Multilayer Perceptron), Traditional RNN (Recurrent Neural Network), LSTM (Long Short-Term Memory) and GRU(Gated Recurrent Units) were evaluated on two synthetic and two real-world time series exhibiting strong chaos, trend, and seasonality. Mackey Glass and Lorenz chaotic time series were simulated in this study to test our DNN models against chaos, while Apple Stock and Melbourne Minimum temperature were two real-world datasets showing increasing trend and seasonality.Experiments demonstrate that GRU based deep learning models outperform all other DNN models in forecasting both real world and synthetic time series
Keywords Time Series Forecasting, Deep Neural Networks, LSTM, GRU, MLP, RNN

1 Introduction

Time series can be defined as a set of data points collected during a period of time which can be daily, weekly, monthly, quarterly or yearly. Time series forecasting which is an active research area that has received indispensable emphasis in various fields like commerce, science and engineering aims to estimate the future by analyzing the past data values. An added temporal component and an underlying non-linearity often associated with real world time series make this entire process very challenging to model so much so that (Keogh and Kasetty; 2002) and (YANG and WU; 2006) rates time series forecasting among the top 10 most challenging problems in machine learning and data mining.

Forecasting and classification are the two essential characteristics of time series modelling. This research focuses on time series forecasting which till now have been primarily achieved using statistical approaches like ARIMA(Autoregressive integrated moving average),SARIMA,Exponential smoothing among others. These statistical techniques are limited by their assumption of the linearity of the underlying time series hence none of

these methods have consistently displayed satisfactory prediction accuracy due to the presence of chaos, noise, seasonality, and non-linearity seen in real-world time series.

Last two decades have seen extensive research on Artificial Neural Networks as an efficient alternative tool for statistical methods in time series forecasting.ANN possess numerous distinctive properties like their nonparametric, nonlinear, data-driven and self-adaptive nature that have made them extremely popular in this domain.More recently though deep learning based neural network methods have generated a considerable amount of interest among researchers to solve time series forecasting problem.Among these deep learning based techniques, LSTM NN(Long Short-Term Memory) (Gers et al.; 1999) (Greff et al.; 2017) which is a type of RNN (Recurrent neural network) with an internal feedback connection and a memory block has shown better performance as compared to traditional forecasting methods.

Lately, new better variants of RNN have been proposed like GRU(gated recurrent unit) (Cho et al.; 2014) which are comparatively less complicated than LSTM in their architectural designs but their performance have not been extensively tested in time series forecasting domain. Also, most conclusions or results about the performance of neural networks in forecasting have been obtained by performing limited empirical studies which have majorly focused on comparing the performance of ANN models with traditions statistical techniques. This study aims to fill this space by conducting an in-depth evaluation of various deep ANN architectures like most popular feed forward NN Multilayer Perceptron, basic RNN, and two of the most advanced RNN architectures LSTM and GRU in time series forecasting.

These DNN architectures were firstly tested on noise-free chaotic synthetic time series like Mackey glass and Lorenz time series.Chaotic time series are the class of synthetic time series that are defined based on recursive , chaotic equations. There results can be considered as an indication of the model's performance before their application against two real-world noisy datasets of Apple stock price and Melbourne City Temperature showing increasing trend and strong seasonal behavior respectively.

Performance of all our models will be compared on MAE(Mean Absolute Error), MSE(Mean Square Error) and and RMSE(Root mean square error) forecasting measures to find the best DNN architecture model in time series forecasting domain. This enhanced understanding regarding strength and shortcomings of each DNN model will help in improving forecasting precision in long run.

2 Scope and Objective

2.1 Scope

The main scope of this research is to empirically evaluate various deep neural network architectures in time series forecasting domain.

2.2 Objective

This work will seek to evaluate forecasting performance of four DNN architectures , Multi-Layer Perceptron (MLP), traditional RNN , LSTM (Long Short-term Memory) and GRU (Gated Recurrent Units) on two simulated noise free chaotic datasets (Mackey Glass and Lorenz Chaos) and two real-world datasets (Apple Stock ,Melbourne Temperature) showing increasing trend and seasonality.

3 Research Question

Which deep neural network architecture is the best fit in time series forecasting domain?

4 Related Work

Time series forecasting as a domain have evolved with time and have seen various techniques being applied on it in various works. This section reviews all this work.

4.1 Traditional Methods of time series forecasting

In their work (Zhang et al.; 2013) lists six types of statistical traditional models that are commonly used for time series forecasting which are Moving Average , AR(Autoregressive), ARIMA ,Seasonal ARIMA and exponential smoothing .The general argument in most of the literature against all of these statistical techniques is regarding their presumption of linearity of the underlying time series.Similar observation is made by (Bandyopadhyay; 2016) in his work where while using ARIMA for predicting gold prices he points out the inability of ARIMA models to detect complex non-linear variations in the data that can be found in most of the real-world time series problems. In their work (Krishnamurthy and Yin; 2002) tried to add non-linearity to the AR models by proposing model that combined hidden Markov model and AR models to forecast non linear time series .However one of the constraints of the proposed model was that its inability to deal with non-stationary time series .Hence the fundamental limitation of traditional approaches in dealing with highly varying and highly non-linear time series motivated researchers to search for alternatives, which led to the increase in popularity of the Artificial Neural networks for forecasting discussed in next section.

4.2 Artificial Neural Network for time series forecasting

In the last decade or so Artificial Neural Networks have gained enormous popularity in time series forecasting and have been widely used in various domains like weather, economic, financial, earthquake etc for forecasting. Artificial neural networks are referred to as universal approximators that do not make any presumptions about linearity and non-linearity of time series and hence can be used to approximate any continuous function with great accuracy. One of the first groundbreaking works in this field was by Chakraborty et al. (1992).This research that eventually opened the door for more future research in incorporating ANN's for forecasting proposed a model for forecasting prices

of various geographic locations using a simple feed-forward neural network that outperformed the traditional ARIMA model with a better RMSE(root mean square error) score.

Unhandled missing and residual values that are often found in real world data can greatly reduce forecasting accuracy. In their research (Chen et al.; 2001) compared the effect of various proportions of missing values on the performance of traditional forecasting model ARIMA and Neural Networks.In this study, Neural Network model achieved better forecasting accuracy as compared to ARIMA models and was found to be less sensitive to missing values then ARIMA models.

Most of the real world time series data would contain two main components which belong to either a seasonal component or a trend. While the trend is a non-linear or linear general systematic component that can change overtime seasonality is a periodically repeating fluctuation that is present in a time series. Detrending and deseasonalization techniques which are normally employed by traditional methods like SARIMA have been criticized by various studies like the one conducted by (Miller and Williams; 2004).The authors of this study believe that these techniques can lead to overestimation. There is a divided opinion among the researcher community regarding the efficiency of Artificial Neural Networks(ANNs) in handling seasonality in a time series. In their research (Nelson et al.; 1999) found that the forecasting accuracy of ANN's trained on seasonal non-adjusted data was lower than those trained on deseasonalized data.Contrary to this view while testing the ability of the most popular type of feed forward neural network Multilayer perceptron in forecasting a time series showing strong seasonality (Adhikari and Agrawal; 2012) concluded that a well designed ANN do not need any pre-adjustments in the data to deal with the strong seasonal component. This perspective was considered in this research while building ANN models on two real-world datasets exhibiting strong trend and seasonality.

Chaotic time series forecasting is another closely related application area in forecasting domain which often can be treated as an indicator of models performance before applying them to the real world (usually business or financial) data forecasting. In their work (Karunasinghe and Liong; 2006) investigated the performance of ANN models on noise-free chaotic Lorenz time series and on noise added chaotic Lorenz time series for MLP ANN model with limited number of parameter choices tried based on trial and error approach. They compared this ANN models predictions accuracy with local prediction models like local polynomial and local averaging models and concluded that in noise-free chaotic time series ANN models show remarkably high accuracy. In another work (Hussein et al.; 2016) employs two chaotic(Lorenz and Mackey glass) and two real world (ACI time series and Sunspot) time series to compare the performance of Elman neural network trained via CCE (cooperative coevolution)algorithm and BPTT(backpropagation through time) algorithms in making single and multi step predictions .In this research CCE clearly outperformed BPTT with lower MAE and RMSE score.

Lately, Deep learning has garnered a lot of attention in researcher community in time series forecasting which is believed to bring next boom of ANN modeling. This is discussed in next section.

4.3 Deep neural networks for time series forecasting:

In recent years, deep learning based methods have generated the lot of curiosity among researchers in time series domain. DNNs with a large number of hidden layers achieves better feature abstraction and hence have the ability to model highly nonlinear real world data with more accuracy and precision. One such study is undertaken by (Gers et al.; 2002) where LSTM (Long Short-Term Memory) based deep ANN model is used in time series forecasting task. LSTM is a better variant of RNN having a special memory cell that gives it an ability to memorize and forget data based on importance and weight of that feature. This study highlighted the superiority of LSTM models in situations where traditional approaches based on time-window fails in dealing with long time lags.Another study undertaken by (Bao et al.; 2017) while using LSTM based deep learner to forecast one step ahead closing prices of six market indices points out at one of the drawbacks of LSTM based models of consuming too much time during their training phase. A better variant of LSTM known as GRU has been proposed recently which because of its comparatively simple structure as compared to LSTM is simpler to implement and compute. This is shown in the research performed by (Fu et al.; 2016) where LSTM and GRU based deep learners are used to predict traffic flow. The results of the experiments performed in this study show GRU NN outperforming LSTM and ARIMA models, having 10 percent lesser MAE score then ARIMA and 5 percent lesser than LSTM. In another work (Kuremoto et al.; 2014) employed deep belief network that was composed of RBM (restricted Boltzman Machine) and MLP(Multi Layer Perceptron) in predicting two chaotic datasets , such as Henon map and Lorez chaos.PSO (particle swarm optimization) was used to decide the network structure. This study concluded that the proposed deep belief network performed lot better than traditional MLP(Multi Layer Perceptron) ANN model.

These researches have shown the promise that deep learning posses in time series forecasting domain but their usability in this domain have not been extensively researched till now. This study seeks to fill up this space by incorporating various deep learning models to forecast a number of synthetic and real-world time series showing strong chaos, trend and seasonality components.

5 Methodology

Time series forecasting with its associated temporal component and various additional dependencies like trends, season, cycle and noise is considered as the unique problem in data mining. In this research, we empirically evaluate the performance of various DNN architectural models in time series forecasting domain to answer our research question of which deep neural network fits best in this domain.

To answer our research question we adopt a two-stage approach.
Stage1:Preliminary tests will be performed to evaluate forecasting performance of all our chosen DNN models on two simulated noise-free datasets showing strong complex chaotic behavior before applying these models to real world datasets.
Stage2:Second stage forecasting tests will be performed on real world noisy datasets showing trend and seasonal components.
The main advantage of this approach is that chaotic time series being less time demanding and less complicated then real world datasets are well suited for preliminary testing

and can be a good indicator of the models quality before its actual application on the real world data. Secondly chaotic time series provides a good base for comparing various models. To build all our DNN models we were motivated to adopt CRISP-DM methodology by the work done by (Jakasa et al.; 2011) where the researchers successfully used this approach to forecasting step ahead spot prices. CRISP-DM is an incremental iterative approach and comprises of six levels as seen in fig 1 that resemble very closely with an Agile methodology.

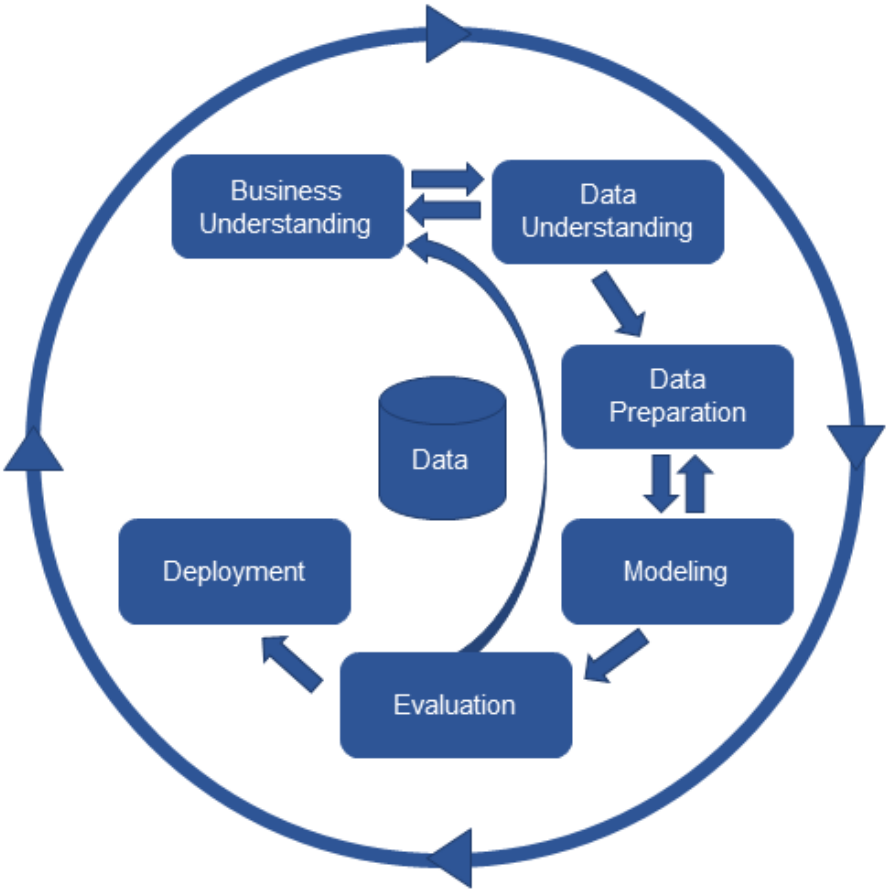

Figure 1: CRISP METHODOLOGY

Data Understanding: In this research, we have employed four-time series data sets. We employed two benchmark chaotic time series Mackey Glass and Lorenz time series which are simulated time series in evaluating the performance of our deep neural network models in time series forecasting. 10000 observations were simulated for both of these benchmark time series using python code. Our remaining two real world time series are the Apple stock prices for the last 5 years i.e. from 2012- 2017 and daily minimum temperature for Melbourne City from 1979 2017.These real world data sets were carefully chosen from different domains, in this case, finance and weather that were showing strong seasonality and trend.

Data Pre-Processing: Data preprocessing or the data wrangling stage is the most time consuming and yet the most crucial stage of the data mining/Analysis project. This

stage basically involved dealing with outliers, missing values, dropping irrelevant features from our datasets and introducing additional features in our datasets if needed.

Modelling: After successful completion of preprocessing stage forecasting models were built using four different deep neural network architecture and in total 16 models were built four models representing our four different datasets on each of the chosen deep neural network architecture. Four different DNN models were selected for this study, MLP(Multilayer Perceptron), traditional RNN and two advancements of RNN which are LSTM(long short-term memory) and GRU(gated recurrent units) .

Evaluation: In this stage performance of all our deep neural network models was evaluated on three popular forecasting measures. Mean Square Error (MSE), Root Mean Square Error (RMSE) and Mean Absolute Error (MAE).

6 Implementation

6.1 Experimental Setup

In order to perform an empirical evaluation of various DNN architectures in time series forecasting domain entire work was arranged in two stages:

Preliminary Stage 1 Tests: In this stage total 8 DNN models were built testing our 4 DNN architectures (MLP,traditional RNN ,LSTM ,GRU) on two synthetic noise free datasets (Mackey Glass and Lorenz Chaos) that were simulated using recursive , chaotic equations in python.These stage is used as an indicator of overall performance of our models before their applications on real world data.

Stage 2 Tests: In this stage another 8 DNN models are built testing our 4 DNN architectures on two real world noisy data showing trend and seasonality. Stage 2 models were more complex in structure as compared to the Stage 1 models having greater number of hidden layers and added neurons to be able to approximate real world data more accurately.

6.2 Data Selection

6.2.1 Synthetic Datasets:

We simulate 10000 observations of our two synthetic datasets the Mackey glass time series dataset and Lorenz time series using python code Both of these synthetic time series are generated using differential equations and both of them show strong chaotic behavior as can be seen in fig 2 and fig 3.

6.2.2 Real World Datasets:

Our remaining two real world time series are the Apple stock prices for the last five years, i.e., from 2012- 2017 and daily minimum temperature for Melbourne City from 1979 2017.These real world data sets were carefully chosen from different domains, in

this case, finance and weather that were showing strong trend and seasonality as can be seen in figure 4 and fig 5.

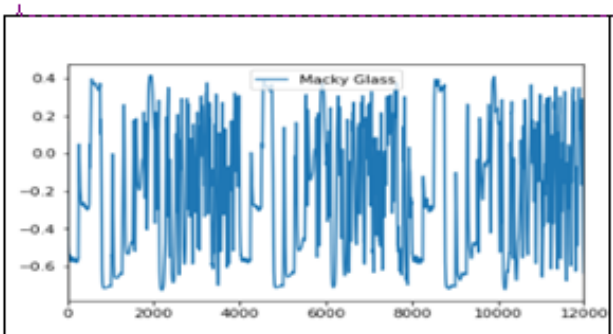

Fig 2: Mackey Glass

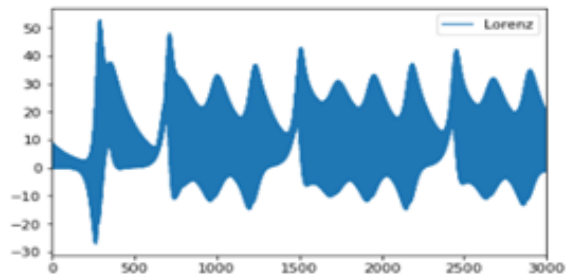

Fig 3. Lorenz Chaotic Time Series

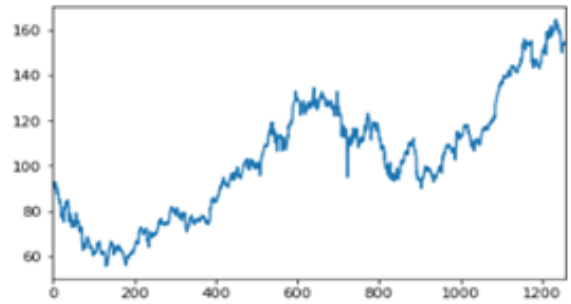

Fig 4: Apple stock data exhibiting increasing trend

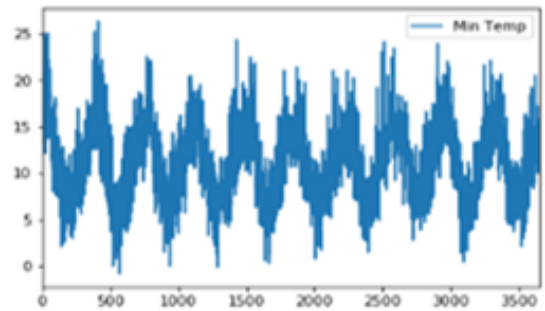

Fig 5 Melbourne minimum temperature data exhibiting seasonality

6.3 Data Division:

To perform our various experiments we divided our data into 60:20:20 split among training, validation and test datasets. Training sets are used to train different models. The Validation dataset is used to performance test ANN models and to avoid ANN models overtraining which occurs when the model starts focusing on the noisy details. Test data is used during ANN model estimation process

6.4 Data Pre-processing

Partial autocorrelation function (PACF) was applied on our data to determine how many past observations are correlated with the current observation that needs to be included in our models which can help improve the accuracy of our models.We used 5-time lags for all our DNN models by looking at the below graph in fig 6 that was generated using statsmod library in python for 30 past observations and here we can see positive correlation exists between x and 5 past observations. All missing values because of time lagging were dropped using pythons dropna() method. For our DNN models, we scaled the data to lie between [-1,1].We then applied log transformation on the target variable y to lie between [0,1] range since deep neural networks build using keras are sensitive towards scale of the input features. Finally, training and test attributes were reshaped into a

suitable format thats accepted by the keras deep learning library. We do not indulge in de-trending and de-seasonalization of the data as a well designed ANN model is capable of handling trends and seasonality without the need for any pre-adjustments Adhikari and Agrawal (2012).

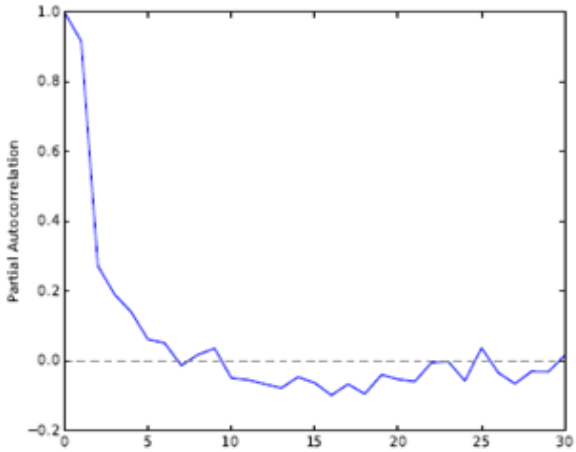

Figure 6: PACF FUNCTION GRAPH

6.5 Framework and Libraries Used:

For building our DNN models we have used Keras deep learning library and have combined it with Theano which is mathematical language as its backend. The full model development is done in python 2.7.For data wrangling, we have used pandas and numpy libraries of python. For scaling and logarithmic transformations, we have used MinMaxScaler and log functions of sklearn library. Pythons' statsmod library is used for performing PACF transformation on our data. Jupyter notebook is used as an IDE for development.

6.6 Hyperparameters Optimization:

Parameters of the model that have to be selected prior to the estimation process are called hyperparameters for e.g., number of hidden layers in the deep neural network, number of neurons in each layer, number of epochs, learning rate etc.

6.6.1 Common Hyperparameters:

Though many hyperparameters can be specific to the ANN model type in this study we have intentionally selected some common hyperparameters for our all our DNN models so that a fair comparison of their performance can be made:

6.5.1.1 Activation Function: Tanh activation function which is a widely known alternative to the sigmoid activation function is chosen as a preferred activation function for all the layers within all our network models except for output layers where linear activation function is used. Tanh function is chosen over sigmoid function since with

its broader range[-1,1] as compared to Sigmoid [0,1] it is found out to be more efficient in modeling highly nonlinear relations (Kalman and Kwasny; 1992) like that normally found in real world time series.

6.5.1.2 Optimizer: The decision concerning the selection of the right optimization algorithm that iteratively updates the weights of the network for the deep neural model can prove to be the difference between getting good results in minutes, hours or days. In this research after experimenting with various available and popular optimizers like the traditional SGD(stochastic gradient decent) , rmsprop ,AdaMax etc we finally selected Adam optimizer for all our models since its computationally more efficient and typically require minimal hyperparameter tuning Kingma and Ba (2014).

6.5.1.3 Epochs: The number of Epoch iterations required for parameter updating by our DNN models was automated by implementing an early call back method in keras.Maximum number of Epochs were kept at 30. While LSTM took the longest time to converge(25-30 epochs), GRU achieved the similar or better results with the lesser number of epochs (19-21) and training time.MLP with no backpropagation were fastest to converge with least number of epochs required generally between 4 to 6.

6.6.2 Model Speciic Hyperparameters:

These parameters were chosen based on trial and error and were specific to the DNN architecture and dataset that was involved.

6.5.2.1 MLP: Multilayer Perceptron that has been analyzed in this paper is the most popular feedforward neural network architecture.First parameter choice that we had to make for our DNN models were the number of hidden layers. In his research, Heaton concludes that for approximating a vast majority of nonlinear functions one hidden layer in a neural network should be enough Heaton (2005). In this research, we explored MLP's up to four hidden layers and finally selected a base MLP model with 2 hidden layers with 4 nodes each, a tanh activation function on hidden layers and linear activation function for output layers for forecasting our synthetic datasets. While forecasting real-world datasets of Apple Stock and Melbourne Temperature we increased the depth of MLP models to 3 hidden layers with 20 neurons each as the problems were more complex to approximate with added noise, trend, and seasonality. This configuration seemed to give us the best accuracy and adding more layers wasn't improving the performance further

6.5.2.2 RNN: Our last deep neural network model designed using Multi layer Perceptron has no sense of time as they dont have the ability to retain information. Unlike this RNNs trained by the technique of backpropagation retains the memory of the previous state and have additional delay units/nodes which introduces a delay between the input and the output and thus helps to retain long-term dependencies in the data. In this research for our synthetic and Apple Stock datasets, RNN model with 1 hidden layers was chosen as it achieved better accuracy than two hidden layer model on these datasets.We increased the depth of RNN models by an additional hidden layer for forecasting noisy and strong seasonal Melbourne Temperature Data. The second parameter that we

experimented with was the number of delay nodes in each layer. After trying models with 2,4,6,8,10 delay node configuration we finally opted for 2 number of delay nodes for simulated noise-free chaotic datasets and 6 number of delay nodes while forecasting real-world datasets Also too many delay nodes in our RNN model were leading to the problem of overfitting.

6.5.2.3 LSTM: LSTM networks with an ability to selectively forget and remember information are considered as a betterment of traditional RNN architectures showing superiority in learning and understanding long-term dependencies in data. For our LSTM models number of hidden layers and delay, nodes were kept similar to our RNN model (1 for synthetic datasets, 2 for real-world datasets) as that configuration gave us best results out of the many we tried. Default Activation functions for the LSTM model were not changed as has been proposed by (Gao and Glowacka; 2016) with hard sigmoid being used as an activation function for inner cells and linear activation function for the outer cells. Batch size which is the number of data observations to train prior to updating weight was kept as 3. Data Shuffle was set to false while building our LSTM models to preserve the temporal order of our data.

6.5.2.4 GRU: GRU an exciting new RNN architecture that was proposed in a work done by (Cho et al.; 2014) in 2014 has fewer parameters in comparison to LSTM but have been found out to deliver same or superior performance than LSTM in various tasks like language modeling ,image recognition etc. For our GRU models all hyperparameters were kept similar to the one we used in our LSTM models as we were getting best results with that set of hyperparameter configurations.

7 Evaluation

There are various forecasting measures that can be used to evaluate and compare the performance of forecasting models. These forecasting measures that are used to estimate the quality of the forecasting model have seemed to mature over the period of time due to numerous forecasting competitions and comparative studies that have been performed in this domain.MSE(mean square error) can be considered as the most popular forecasting but as pointed out by (Armstrong and Collopy; 1993) it cannot be used alone for making comparisons between various time series, so in this study we have not relied on only one forecasting measure but have used a combination of three measures, namely MAE(mean absolute error), MSE(mean square error) AND RMSE(root mean square error).

7.1 Experiment / Case Study 1

In the first experimental study, we compare the performance of different ANN architectures i.e. MLP, RNN, LSTM, and GRU on two benchmark synthetic time series Mackey Glass and Lorenz time series that are known for their chaotic behavior. We simulate 10000 observations of each of these time series to measure our ANN models on chaotic time series

7.2 Experiment / Case Study 2

In our second experiment, we tested our ANN models on a real world financial data set of Apple Stock that is showing an increasing trend as can be seen in fig 3

7.3 Experiment / Case Study 3

In our third experiment we tested performance of our ANN models on a real world data set from weather forecasting domain containing daily minimum temperature recorded for Melbourne City from 1979-2017.This data was showing strong seasonality as can be seen in fig 3 .

7.4 Discussion

As can be seen in fig 8,9,10,11,12 all DNN models performed reasonably well on synthetic Macky Glass and Lorenz time series with new RNN advancements LSTM and GRU outperforming other two ANN architectures with slightly better MSE , RMSE and MAE scores which can be attributed to their inbuilt architectural design that is suited to understand long-term dependencies in data.GRU in particular performed extremely well in forecasting the chaotic behavior of time series as can be seen in fig 11 and was a clear winner in predicting chaos with a very low MSE ,RMSE and MAE scores of 0.63,4.80 and 4.40 respectively on Macky Glass time series and MSE ,RMSE and MAE scores of 0.45,3,50 and 3.14 respectively on Lorenz time series as seen in fig 7.MLP models though achieved reasonable scores for forecasting chaotic behavior but it has to be noted that MLP models employed in our research were more complex in structure as compared to traditional RNN, LSTM, and GRU models.MLP models needed 2 hidden layers with 4 neurons each to compete with RNN models traditional RNN ,LSTM and GRU models that needed only one layer with 2 delay units to approximate chaotic behavior with high accuracy.High forecasting accuracy achieved by our RNN models and deep MLP model on Mackey Glass and Lorenz time series can also be attributed to the fact that these were simulated time series that had no noise element in them, unlike real world time series.

MACKY GLASS TIME SERIES			
	MSE	RMSE	MAE
MLP	1.37%	9.78%	10.17%
SimpleRNN	1.43%	6.98%	6.75%
LSTM	0.68%	5.96%	4.52%
GRU	0.63%	4.80%	4.42%

LORENZ CHAOTIC TIME SERIES			
	MSE	RMSE	MAE
MLP	0.91%	7.78%	7.78%
SimpleRNN	0.69%	5.98%	5.92%
LSTM	0.46%	4.01%	3.98%
GRU	0.45%	3.50%	3.14%

APPLE STOCK TIME SERIES			
	MSE	RMSE	MAE
MLP	14.76%	28.56%	28.56%
SimpleRNN	16.52%	28.12%	27.70%
LSTM	4.55%	15.05%	14.44%
GRU	1.24%	7.18%	7.09%

MELBOURNE CITY TEMPERATURE TIME SERIES			
	MSE	RMSE	MAE
MLP	4.10%	15.18%	15.16%
SimpleRNN	3.90%	14.65%	14.40%
LSTM	3.90%	14.65%	14.40%
GRU	3.90%	14.65%	14.40%

Figure 7: FORECASTING MEASURES

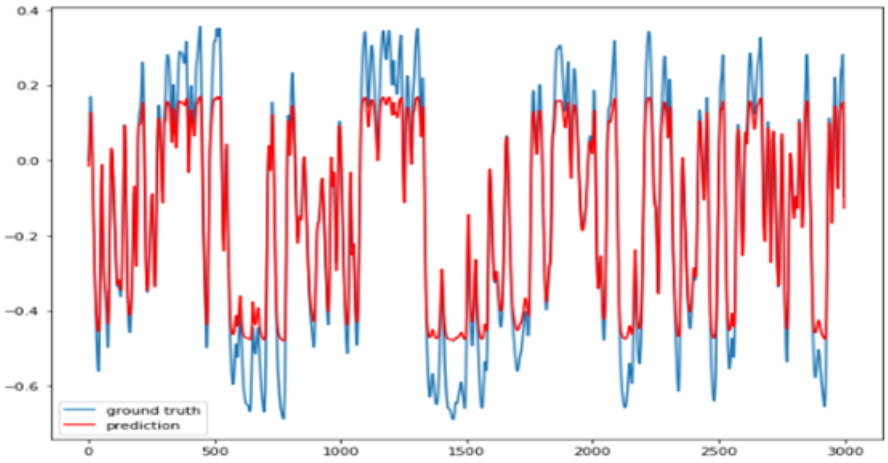

Figure 8: MLP MACKEY GLASS FORECASTING

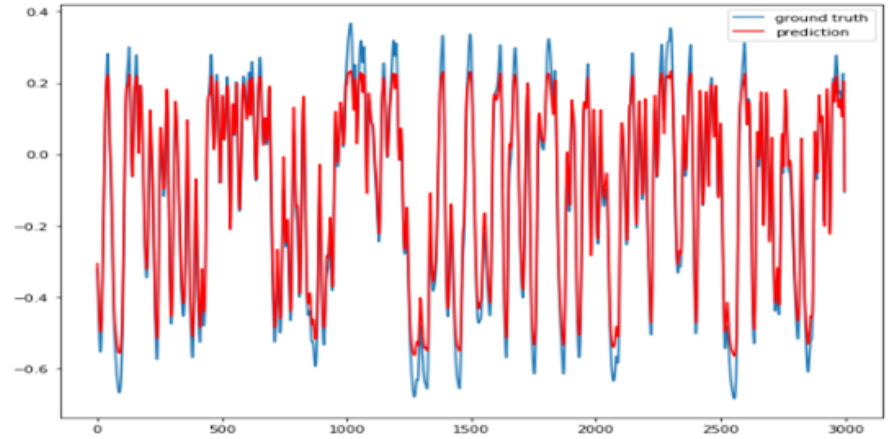

Figure 9: RNN MACKEY GLASS FORECASTING

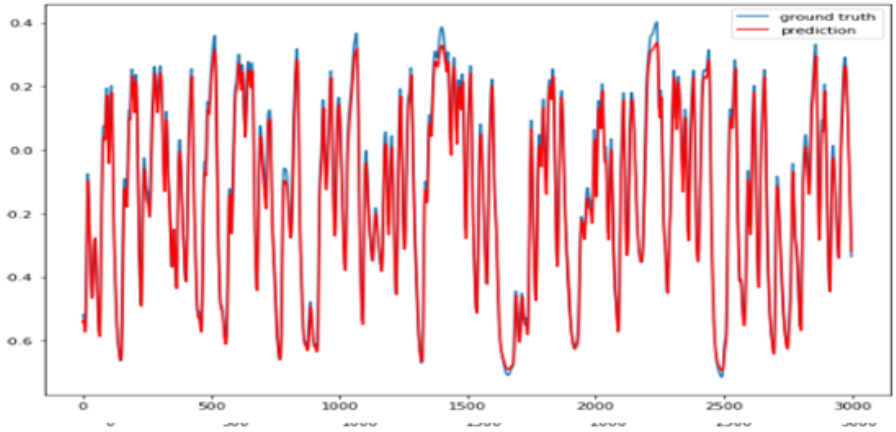

Figure 10: LSTM MACKEY GLASS FORECASTING

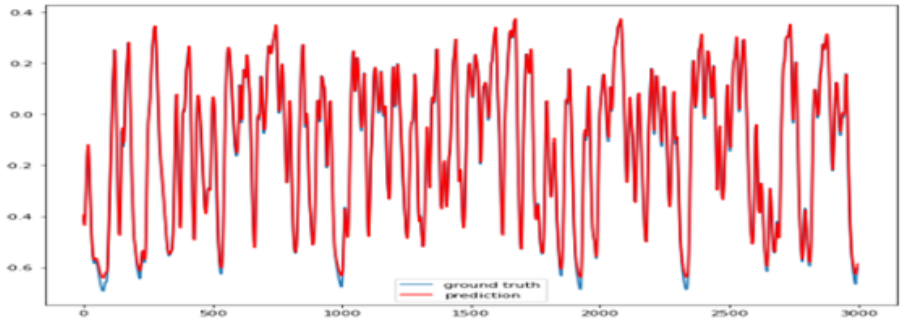

Figure 11: GRU MACKEY GLASS FORECASTING

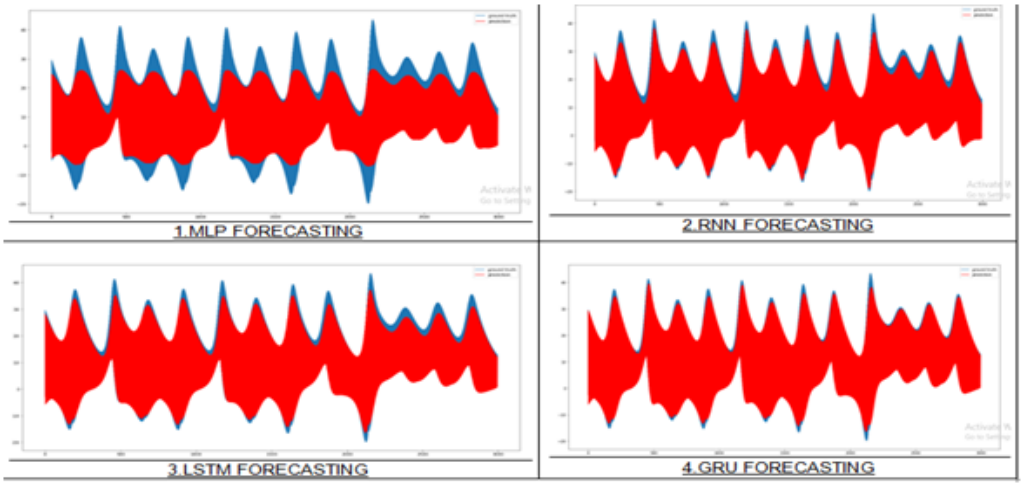

Figure 12: Lorenz Chaotic Series Forecasting

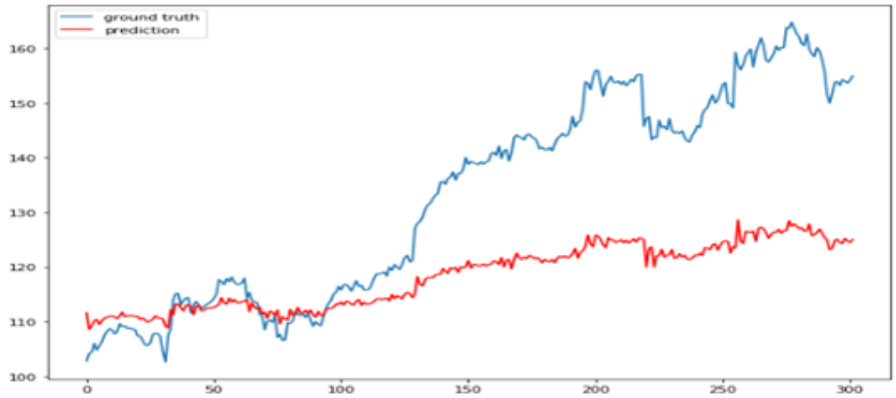

Figure 13: MLP FORECASTING APPLE

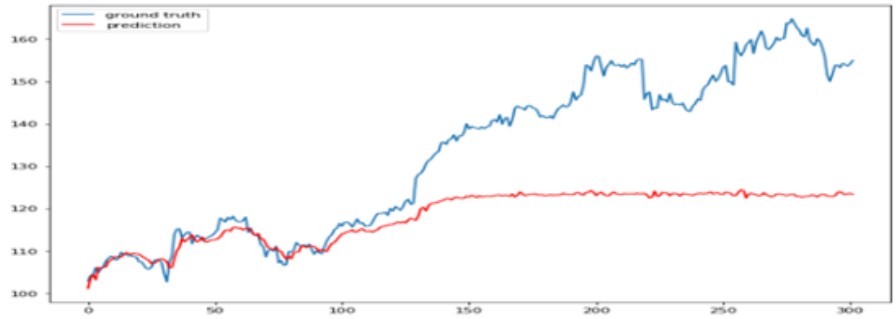

Figure 14: RNN FORECASTING APPLE STOCKS

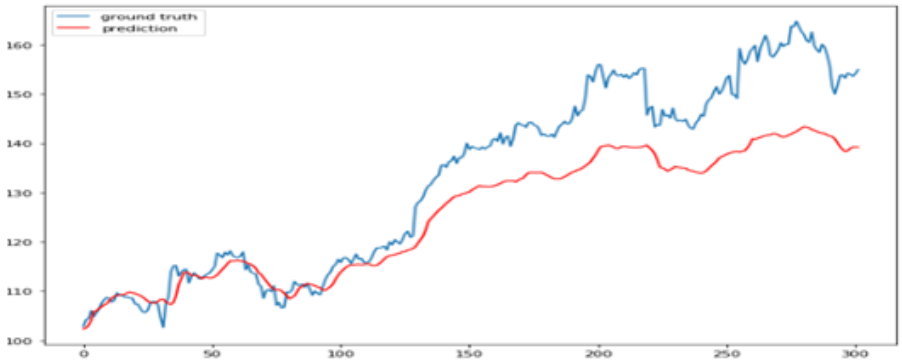

Figure 15: LSTM FORECASTING APPLE STOCKS

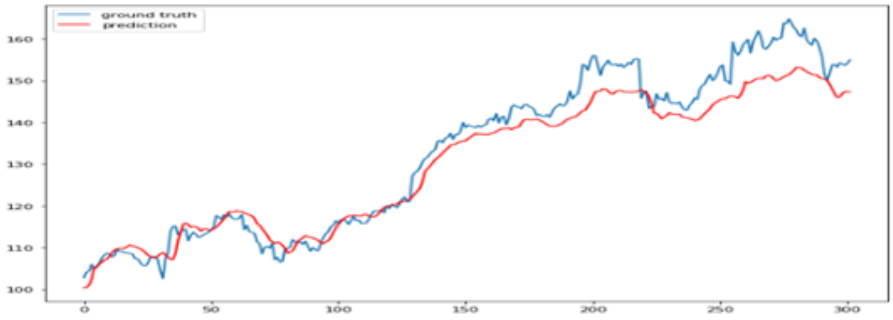

Figure 16: GRU FORECASTING APPLE STOCKS

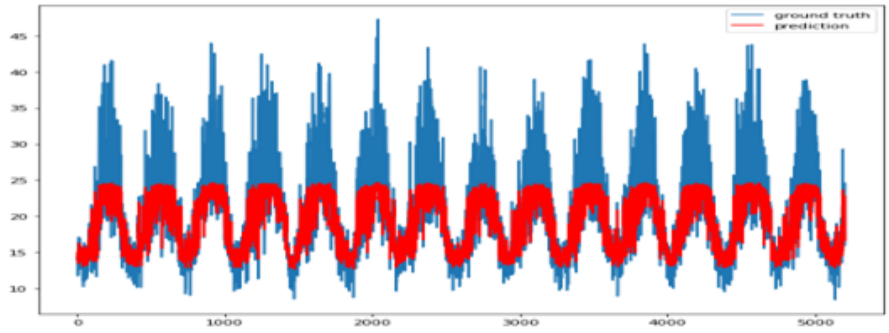

Figure 17: MLP FORECASTING MELBOURNE TEMPERATURE

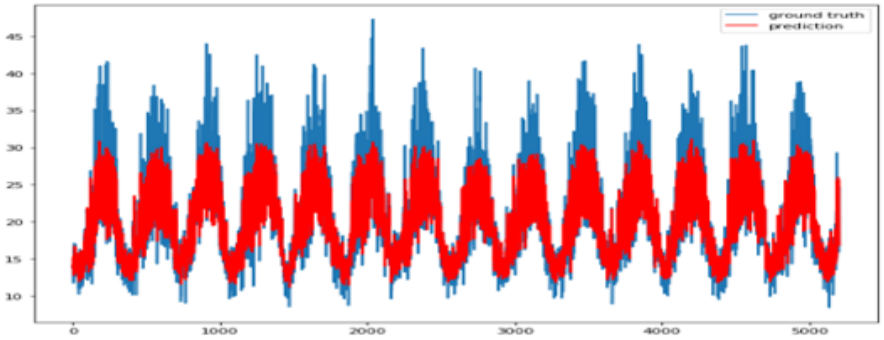

Figure 18: RNN FORECASTING MELBOURNE TEMPERATURE

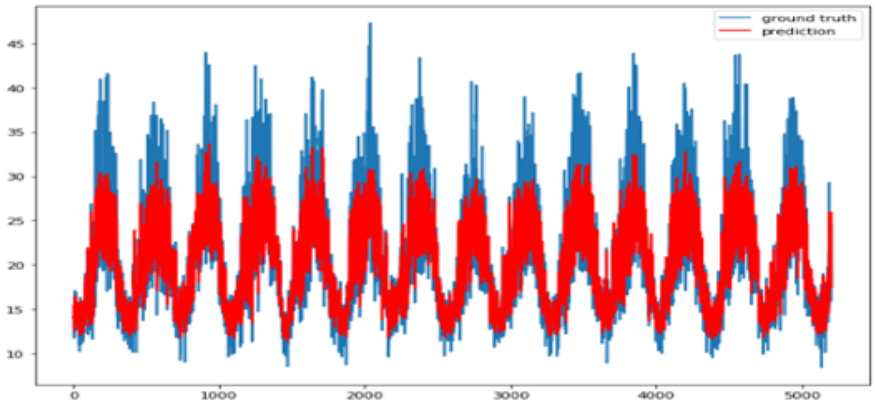

Figure 19: LSTM FORECASTING MELBOURNE TEMPERATURE

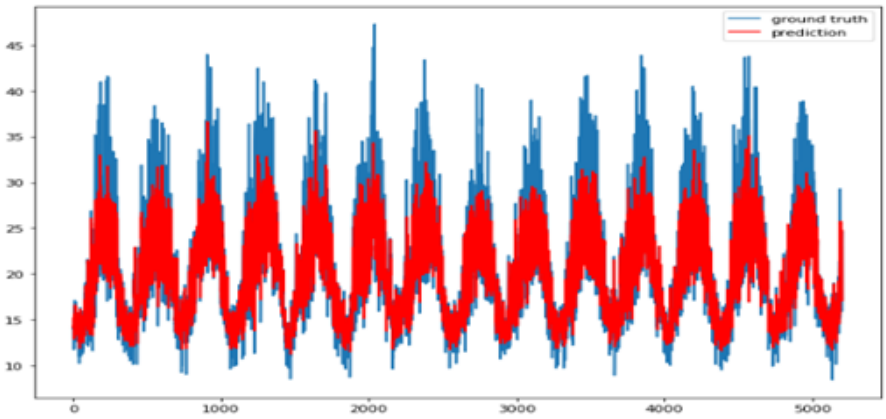

Figure 20: GRU FORECASTING MELBOURNE TEMPERATURE

While forecasting real-world time series data having noise, trend and seasonality GRU model again outperformed all other ANN models (fig 16 and fig 20) .Their performance was particularly impressive with Apple Stock price forecasting .This problem had an additional increasing trend component in it and GRU based DNN models achieved lowest MSE, RMSE and MAE score of 1.24,7.18 and 7.19 respectively in this problem. While forecasting Melbourne minimum temperature even though their MSE , RMSE and MAE

score was tied with RNN and LSTM's scores at 3.90,14.65 and 14.40 respectively, GRU models can be seen understanding the highs and lows of the seasonal data slightly better than the other two as can be seen above in fig 7 and fig 20. MLP performed worst among 4 in both the cases as evident from their high forecasting scores in fig 7 and time series graphs in fig.13 and 17. While looking at graph in figure 13 we can say that MLP models even with 2 hidden layers with 20 nodes each and no backpropagation failed to approximate the increasing apple stock trend with satisfactory accuracy and achieved MSE ,RMSE and MAE scores of 14.76,28.56 and 28.5 respectively.

8 Conclusion and Future Work

In this research we empirically evaluated the forecasting performance of various deep neural network architectures in time series forecasting domain. We reviewed four different DNN architectures ,MLP (multilayer perceptron) which is the most popular feedforward neural network architecture, RNN(recurrent neural network) and its two latest advancements LSTM(long short term memory) and GRU(gated recurrent units) in time series forecasting.

To get a more robust and general conclusion about the performance of our DNN models we first evaluated them on Mackey Glass and Lorenz time series which are two noise-free simulated datasets that are known to show complex chaotic dynamics. We followed this up with analyzing the performance of these models on two real world noisy datasets from financial(Apple stock) and weather forecasting domain(Melbourne Temperature) showing increasing trend and seasonality components respectively. Among the numerous hyper-parameter choices that were available, trial and error method was followed which lead us to the final DNN models parameter choice.

Following important conclusions can be drawn from our experiments.
GRU which is the latest advancement in RNN architecture model was the clear winner in forecasting both simulated and real world datasets that were showing chaos, trend and seasonality and to answer our research question was the best fit DNN model in forecasting performed in our study. In forecasting noise-free simulated chaotic time series all our DNN models showed high forecasting accuracy but we had to add more complexity to our feedforward MLP model for it to be able to compete with other RNN models (tradiotnal RNN, LSTM, GRU).All our RNN models only needed one hidden layer with 2 delay nodes to approximate chaotic behavior with high accuracy showing their strong temporal sequence learning strength as compared to 2 hidden layers with 4 neurons each needed by our MLP model. GRU with lowest MSE, MAE and RMSE scores outperformed all other models in this problem even though difference in accuracy among these models wasn't huge for this problem.In forecasting Apple stock price which was our first real world dataset showing an increasing trend component GRU with its advanced gated mechanism clearly showed its superiority over remaining three DNN models achieving outstanding MSE, MAE and RMSE scores.In this case, study our much more complex MLP which is our only feed forward model with 2 hidden layers having 20 neurons each was outperformed by much more simpler GRU model having only one hidden layer with 4 delay nodes. While forecasting a strong seasonal noisy Melbourne minimum temperature time series even though GRU did not perform significantly better then the LSTM

and simple RNN models but it was comparatively better in forecasting highs and lows of this time series more accurately than the other two. Another important aspect that was highlighted by this case study in particular was that even though gated RNN models (GRU AND LSTM) did not outperform traditional RNN model in this study but they needed lesser number of delay units 5 in each layer when compared with our RNN model which needed 6 in each layer to approximate this seasonal non linearity.MLP even with an additional one hidden layer as compared to RNN models was the worst performing among 4 models.

We followed the trial and run approach to choose our final set of optimal hyperparameters to train our DNN models on. Even though we implemented early stopping mechanism to automate the right number of epochs runs required by our models to prevent overfitting in future work optimization of the model hyperparameters can be achieved by implementing Bayesian optimization using Hyperas library that would choose the optimal model parameters without the need of following trial and run approach that can improve the forecasting accuracy further and save model development time. Also, latest proposals in RNN architecture designs like MGU (Zhou et al.; 2016) and GF-RNN (Chung et al.; 2015) can also be incorporated in this study.

.

References

Adhikari, R. and Agrawal, R. (2012). Forecasting strong seasonal time series with artificial neural networks, *Journal of Scientific and Industrial Research* **71**: 657–666.

Armstrong, J. and Collopy, F. (1993). Error measures for generalizing about forecasting methods: Empirical comparisons, *Long Range Planning* **26**(1): 150.

Bandyopadhyay, G. (2016). Gold price forecasting using arima model, *Journal of Advanced Management Science* pp. 117–121.

Bao, W., Yue, J. and Rao, Y. (2017). A deep learning framework for financial time series using stacked autoencoders and long-short term memory, *PLOS ONE* **12**(7): e0180944.

Chakraborty, K., Mehrotra, K., Mohan, C. K. and Ranka, S. (1992). Forecasting the behavior of multivariate time series using neural networks, *Neural Networks* **5**(6): 961–970.

Chen, H., Grant-Muller, S., Mussone, L. and Montgomery, F. (2001). A study of hybrid neural network approaches and the effects of missing data on traffic forecasting, *Neural Computing Applications* **10**(3): 277–286.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation, *arXiv preprint arXiv:1406.1078* .

Chung, J., Gulcehre, C., Cho, K. and Bengio, Y. (2015). Gated feedback recurrent neural networks, *International Conference on Machine Learning*, pp. 2067–2075.

Fu, R., Zhang, Z. and Li, L. (2016). Using lstm and gru neural network methods for traffic flow prediction, *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)* .

Gao, Y. and Glowacka, D. (2016). Deep gate recurrent neural network, *Asian Conference on Machine Learning*, pp. 350–365.

Gers, F. A., Eck, D. and Schmidhuber, J. (2002). Applying lstm to time series predictable through time-window approaches, *Neural Nets WIRN Vietri-01*, Springer, pp. 193–200.

Gers, F. A., Schmidhuber, J. and Cummins, F. (1999). Learning to forget: Continual prediction with lstm.

Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R. and Schmidhuber, J. (2017). Lstm: A search space odyssey, *IEEE transactions on neural networks and learning systems* .

Heaton, J. (2005). Understanding the kohonen neural network. introduction to neural networks with java. heaton research.

Hussein, S., Chandra, R. and Sharma, A. (2016). Multi-step-ahead chaotic time series prediction using coevolutionary recurrent neural networks, *Evolutionary Computation (CEC), 2016 IEEE Congress on*, IEEE, pp. 3084–3091.

Jakasa, T., Androcec, I. and Sprcic, P. (2011). Electricity price forecasting x2014; arima model approach, *2011 8th International Conference on the European Energy Market (EEM)* .

Kalman, B. L. and Kwasny, S. C. (1992). Why tanh: choosing a sigmoidal function, *Neural Networks, 1992. IJCNN., International Joint Conference on*, Vol. 4, IEEE, pp. 578–581.

Karunasinghe, D. S. and Liong, S.-Y. (2006). Chaotic time series prediction with a global model: Artificial neural network, *Journal of Hydrology* **323**(1): 92–105.

Keogh, E. and Kasetty, S. (2002). On the need for time series data mining benchmarks, *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '02* .

Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization, *arXiv preprint arXiv:1412.6980* .

Krishnamurthy, V. and Yin, G. G. (2002). Recursive algorithms for estimation of hidden markov models and autoregressive models with markov regime, *IEEE Transactions on Information Theory* **48**(2): 458–476.

Kuremoto, T., Obayashi, M., Kobayashi, K., Hirata, T. and Mabu, S. (2014). Forecast chaotic time series data by dbns, *Image and Signal Processing (CISP), 2014 7th International Congress on*, IEEE, pp. 1130–1135.

Miller, D. M. and Williams, D. (2004). Damping seasonal factors: Shrinkage estimators for the x-12-arima program, *International Journal of Forecasting* **20**(4): 529–549.

Nelson, M., Hill, T., Remus, W. and O'Connor, M. (1999). Time series forecasting using neural networks: Should the data be deseasonalized first?, *Journal of forecasting* **18**(5): 359–367.

YANG, Q. and WU, X. (2006). 10 challenging problems in data mining research, *International Journal of Information Technology Decision Making* **05**(04): 597–604.

Zhang, X., Liu, Y., Yang, M., Zhang, T., Young, A. A. and Li, X. (2013). Comparative study of four time series methods in forecasting typhoid fever incidence in china, *PLoS ONE* **8**(5): e63116.

Zhou, G.-B., Wu, J., Zhang, C.-L. and Zhou, Z.-H. (2016). Minimal gated unit for recurrent neural networks, *International Journal of Automation and Computing* **13**(3): 226–234.