Capstone Project: Final Manuscript

**Can Neural Networks outperform Linear Regression in Stock Market Prediction?**

*Mohamed Mohamar*

*Data Science Institute, Saint Peter's University*

[*mmohamar@saintpeters.edu*](mailto:mmohamar@saintpeters.edu)

**Abstract**

The ferocious competitive fight between stock market actors have never been more intense than today. Corporations have invested millions of dollars to build state of the art powerful hardware and software for the sole purpose of analyzing and understanding stock market fluctuation patterns, trends, and risks. Yet, their margin of gain is still in cents. With everyone having access to the same historical data, it is not certain if anyone stock market professional can have any advantage over other actors. Lots of machine learning models have been conceived over years, including linear regression models. Over time some produced better results than others. As more computing power becomes available in recent years, other more complex models have been look at. Among those complex models is Neural Networks. The exploration of neural networks began decades ago but, they have been some kind of black box for us for very long. However, they are nowadays more understood and have application in multiple disciplines including data analysis and simulation. R programming packages LM and NEURALNET are used to perform a comparative analysis of linear regression and neural network time series analysis to build a model to predict and forecast stock market volatility. Results show that Neural Networks model outperform the linear regression model.

**Work by Competitors**

A lot of great work has been done in the field of Data Science using Neural Networks, applied to varieties of disciplines of humanity. Following are four of the recent state of the art research papers done about this exciting and challenging field of data science, where the authors utilized Neural Networks or a variation or hybrid of Neural Networks.

1. ***An adaptive local linear optimized radial basis functional neural network model for financial time series prediction***

In this paper, A. Patra, S. Das, S. N. Misha, and M. R. Senapati apply a “local linear radial basis functional neural network (LLRBFNN) model” to classify Yahoo Inc. financial data. According to the authors, “The LLRBFNN model is learned by using the hybrid technique of backpropagation and recursive least square algorithm. It uses a local linear model in between the hidden layer and the output layer in contrast to the weights connected from hidden layer to output layer in typical neural network models.”  The authors show that “the obtained prediction result is compared with multilayer perceptron and radial basis functional neural network with the parameters being trained by gradient descent learning method. The proposed technique provides a lower mean squared error and thus can be considered as superior to other models. The technique is also tested on linear data, i.e., diabetic data, to confirm the validity of the result obtained from the experiment.” (*Patra et Al. 2017)*

1. ***Forecasting stock market indices using hybrid network***

Here are how the authors present their work: “In this paper, a hybrid network consisting of a trigonometric Functional Link Artificial Neural Network (FLANN) and Fuzzy Logic System named as Functional Link Neural Fuzzy (FLNF) Model is used to predict the stock market indices. The proposed model uses a functional link neural network to the consequent part of the fuzzy rules. The consequent part of FLNF model is a non-linear combination of input variables. Two stock market indices (data sets) i.e., Bombay Stock Exchange and Standard's and Poor's (S&P500) are collected for experimentation. Samples for 4000 trading days from 1st March 1993 to 23rd July 2009 are collected from the former and 3228 trading days from 1st March 1993 to 09th June 2006 for the later. This model is used to forecast stock market indices one day, one week and one month in advance. A comparative analysis between the proposed hybrid model and that of FLANN has also been given.” (*Satapathy et Al. 2015)*

1. ***Does Artificial Neural Network Forecast Better For Excessively Volatile Currency Pairs.***

The authors present their work as following: “This study predicts the exchange rates for three currency pairs (USD-INR, GBP-INR, and EUR-INR). We have used multi-layer perceptron (MLP) neural network architecture based on feed-forward with back-propagation learning method. The sample of the study covers daily data for the period from January 2009 to January 2016. The findings of the study confirm that the neural network predicts better for more volatile currency pairs (GBP-INR and EUR-INR) as compared to a less volatile currency pair (USD-INR). The study further observes that the optimal forecast horizon for the neural network model should be equal to the optimal lag length used in the construction of the model. This study aims to contribute in the area of foreign exchange forecasting. Exchange rate plays a crucial role in the macro-economy of a country. Hence, prediction of currency exchange rate becomes imperative for various stakeholders such as government, the central bank, and investors to maximize the returns and minimize the risk in their decision-making.” (*Kumar et Al. 2016)*

1. ***Single-hidden layer neural networks for forecasting intermittent demand***

In this article, the authors are proposing a method that uses a single-hidden layer feedforward neural network for forecasting intermittent demand. This is based on the back-propagation gradient-descent, perceptron algorithm described in the algorithm overview section. However, due to some of the drawbacks of back-propagation gradient descent algorithms including, slow convergence, setting learning parameters, etc. They are relying on a “faster learning algorithm”, first introduced by *Huang et al. (2006),* and later reviewed by the same authors *Huang et al. (2015).* Their proposal aims to “helping the network to learn the temporal behavior of the time series in terms of zero/non-zero demand” *(Lollia et al. 2017).* They compared the accuracy of their forecasting model with other Neural Networks that dealt with intermittent demands *(Guttierez et al., 2008; Mukhopadhyay et al. 2012; Croston, 1972; Syntetos and Boylan, 2005)*

As conclusion, the authors argue that “this comparison was then enriched by adopting two different accuracy metrics on different time horizons. Such a detailed comparison aims at bridging the gap between theory and practice of ANNs in the field of intermittent demand. *(Lollia et al. 2017)*. In fact, the potential for implementation of ANNs in real environments can only increase by providing useful guidelines about their design and training for practitioners. Finally, a statistical analysis of the networks’ performance was conducted, for robust validation of the results.” *(Lollia et al. 2017)*

1. **Multi-step ahead electricity price forecasting using a hybrid model based on two-layer decomposition technique and BP neural network optimized by firefly algorithm**

“In the deregulated competitive electricity market, the price which reflects the relationship between electricity supply and demand is one of the most important elements, making it crucial for all market participants to precisely forecast the electricity price. However, electricity price series usually has complex features such as non-linearity, non-stationarity and volatility, which makes the price forecasting turn out to be very difficult. In order to improve the accuracy of electricity price forecasting, this paper first proposes a two-layer decomposition technique and then develops a hybrid model based on fast ensemble empirical mode decomposition (FEEMD), variational mode decomposition (VMD) and back propagation (BP) neural network optimized by firefly algorithm (FA). The proposed model is unique in the sense that VMD is specifically applied to further decompose the high frequency intrinsic mode functions (IMFs) generated by FEEMD into a number of modes in order to improve the forecast accuracy. To validate the effectiveness and accuracy of the proposed model, three electricity price series respectively collected from the real-world electricity markets of Australia and France are adopted to conduct the empirical study. The results indicate that the proposed model outperforms the other considered models over horizons of one-step, two-step, four-step and six-step ahead forecasting, which shows that the proposed model has superior performances for both one-step and multi-step ahead forecasting of electricity price.” *(Wanga et Al. 2017)*

1. **A novel macroeconomic forecasting model based on revised multimedia assisted BP neural network model and ant Colony algorithm**

“In this paper, we propose a novel macroeconomic forecasting model based on the revised multimedia assisted BP neural network model and the ant colony algorithm. Macroeconomic forecasting foundation forecasts the object past and present operating law, therefore, when the operational predict that must describe the analysis and this rule. Because the limitation and forecast technique of choice fault forecast technique can create the uncertainty of the forecasting result, our model mainly focus on the following two aspects. (1) Uncertainty in forecasting method selection errors is even more evident. The probability that the wrong prediction method brings the correct prediction result is very small. (2) Limitations of the forecasting methods. Any kind of forecasting method has its applicable conditions and the environment, it is not omnipotent, nor is it immutable, therefore, more of the state-of-the-art techniques should be researched to enhance the traditional approaches. We use the ant colony algorithm to modify the BP model to make it fit for holding the character that forecasting that a point refers to forecasting a definite value, this value and actual value completely same possibility is very low, this explained that a point forecast successful probability is very low, therefore uses the forecasting result judgement forecast method the fit and unfit quality to be not very comprehensive. Forecast that a sector refers to the future reality leaving in the prediction interval, or prediction interval including the future realistic value which will hold special meaning. The experiment on the stock, gold, exchange and inflation indicate that the proposed model can predict the price well with the satisfactory result.” *Kuang et Al. 2017)*

1. **Stock Market Prediction by Non-Linear Combination based on Support Vector Machine Regression Model**

“Stock market predictions comprise challenging applications of modern time series forecasting and are essential to the success of many businesses and financial institutions. In this paper, stock market forecasting is based on Support Vector Machine (SVM) regression. Firstly, using different linear regression model to extract linear characteristics of stock market system. Secondly, using different Neural Network algorithms to extract nonlinear characteristics of stock market system. Finally, the SVM regression is used for the nonlinear combination forecasting model of different stock exchange prices. Empirical results obtained reveal that the prediction by using the nonlinear combination model is generally better than those obtained using other models presented in this study in terms of the same evaluation measurements. Those results show that that the proposed nonlinear modeling technique is a very promising approach to financial time series forecasting.” (*Kumar et Al. 2016)*

1. **Times Series Forecasting using Chebyshev Functions based Locally Recurrent neuro-Fuzzy Information System**

“The model proposed in this paper, is a hybridization of fuzzy neural network (FNN) and a functional link neural system for time series data prediction. The TSK-type feedforward fuzzy neural network does not take the full advantage of the use of the fuzzy rule base in accurate input-output mapping and hence a hybrid model is developed using the Chebyshev polynomial functions to construct the consequent part of the fuzzy rules. The model to be known as locally recurrent neuro fuzzy information system (LRNFIS) is used to provide an expanded nonlinear transformation to the input space thereby increasing its dimension which will be adequate to capture the nonlinearities and chaotic variations in the time series. The locally recurrent nodes will provide feedback connections between outputs and inputs allowing signal flow in both forward and backward directions, giving the network a dynamic memory useful to mimic dynamic systems. For training the proposed LRNFIS, an improved firefly-harmony search (IFFHS) learning algorithm is used to estimate the parameters of the consequent part and feedback loop parameters. Three real world time series databases like the electricity price of PJM electricity market, the widely studied currency exchange rates between US Dollar (USD) and other four currencies i.e. Australian Dollar (AUD), Swiss Franc (CHF), Mexican Peso (MXN), Brazilian Real (BRL), along with S&P 500 and Nikkei 225 stock market data are used for performance validation of the newly proposed LRNFIS*.” (Parida et Al. 2017)*

**Contribution**

This work is an effort to contribute to the research and the exploration of Neural Networks. The contribution is to help understand Neural Networks better and show how they can be successfully applied to stock market data analysis. The proposal is inspired by the previous excellent work that has been done by data scientists and other professionals using Neural Networks. My idea is to use R programming packages LM and NEURAL NETWORK to perform a comparative analysis of linear regression and neural network time series analysis to build a model to predict and forecast stock market volatility. The ultimately goal is to build a high performant stock market analysis compact model that can be used as an investment tool.

**Data**

The original full data set that I am using was provided by Professor Robert Finn last semester during our DS-640 course. It is a stock market data set. It contains data about Income Statement, Cash Flow Statement, Balance Sheet, and Metrics and Ratios.

I am only using a subset of the data corresponding to the ARQ (As Reported Quarterly) listings, meaning where Dimension = ARQ. The Rows are the quarterly calendar date Time Series. The Columns are twenty chosen indicators along with the calculated Returns and Log of returns.

My list of twenty chosen indicators or factors is the following:

1. CURRENTRATIO Current Ratio ARQ, MRQ, ARY, MRY
2. DE Debt to Equity Ratio ARQ, MRQ, ARY, MRY
3. DIVYIELD Dividend Yield
4. FCFPS Free Cash Flow per Share ARY,MRQ, ART, MRT,

MRQ, MRY

1. GROSSMARGIN Gross Margin ART, MRT
2. PAYOUTRATIO Payout Ratio ART, MRT
3. SPS Sales per Share ART, MRT
4. NETMARGIN Profit Margin ART, MRT
5. BVPS Book Value per Share ARQ, MRQ, ARY, MRY
6. EVEBITDA Enterprise Value over EBITDA ART, MRT
7. PE1 Price to Earnings Ratio ART, MRT
8. PS1 Price to Sales Ratio ART, MRT
9. TBVPS Tangible Asset Book Value per Share ARQ, MRQ,

ARY, MRY

1. PRICE Share Price (Adjusted Close)
2. PS Price Sales Damodaran Method ART, MRT
3. EBITDAMARGIN EBITDA Margin ART, MRT
4. PB Price to Book Value ARQ, MRQ, ARY, MRY
5. REVENUE Revenues
6. EPS Earnings per Basic Share
7. PE Price Earnings Damodaran Method ART, MRT

**Algorithm Overview**

Considering the linear regression modelthe functions: for *i = 1… d* in the expression: =,, …, are called *basis functions.* Now*,* instead of using linear regression, let’s allow the functions of the independent variables, or basis functions, to be adaptive.This basically means that we use parametric forms for the basis functions in which the parameter values are adapted during training. A very successful application of this model is *the feed-forward neural network*, also known as the multilayer perceptron, which is based on the *perceptron algorithm (PA).* PA is similar to the linear model but, with a step activation function:

Where *f (a) =*

The algorithm trains using the perceptron criterion, with a form of:

() = −, where M is the set of misclassified patterns.

We are looking for a weight vector such that

, and .

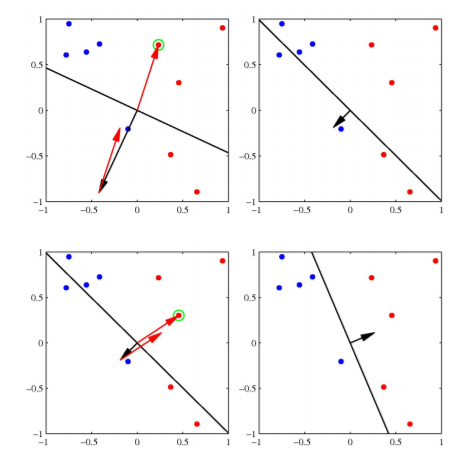
It also uses a stochastic gradient descent:

-

Now, since the perceptron function is unchanged if is multiplied by a constant, we may set = 1. We deduct the following expression:

*The perceptron convergence theorem says: If there exists an exact solution, i.e. if the training set is indeed linearly separable, then the Perceptron Algorithm will find a solution in finite number of steps.*

Following is the geometry of the perceptron algorithm:



As comparison, many applications, neural networks can be significantly more compact and faster to evaluate than a support vector machine. But this property of neural networks comes at a certain price which is the following: the likelihood function, which forms the basis for network training, is no longer a convex function of the model parameters. However, in practice, it is often worth investing substantial computational resources during the training phase in order to obtain a compact model that is fast at processing new data sets.

Linear models for regression and classification are based on linear combinations of fixed nonlinear basis functions of the form of:

*Y (****X***,) = *f*

Where *f* is a nonlinear activation function in the case of classification and is the identity in the case of regression.

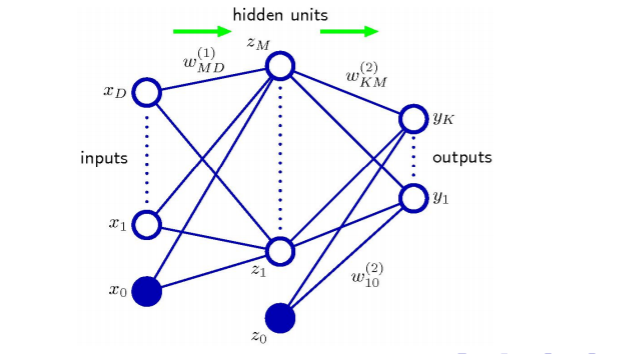
We extend this model by making the basis functions depend on parameters and then allow these parameters to be adjusted, along with the coefficients {}, during training. Hence, the basic neural network model can be described a series of functional transformations.

First we construct *M* linear combinations of the input variables … in the form of: = + where *j = 1 . . . M* and (1) indicates that the corresponding parameters are in the first layer of the network. The parameters are referred to as *weights*, the parameters as *biases*, and the quantities are known as *activations*. Each of them is then transformed using a differentiable, nonlinear activation function *h* to give *= h* ().

These quantities correspond to the outputs of the basis functions and in the context of neural networks are called *hidden* units. The nonlinear functions *h* are generally chosen to be sigmoidal functions such as the logistic sigmoid or the arc tan function. These values are again linearly combined to give output *unit activations:* = + where *k = 1 . . . K* and *K* is the total number of outputs.

This transformation corresponds to the second (2) layer of the network, and again the entities are bias parameters.

Finally, the output unit activations are transformed using an appropriate activation function to give a set of network outputs noted as.



A common choice for an activation function is the following:

= σ ( ), where σ (*a*) =

Finally, we can combine all these various stages to give the overall network functions that, for sigmoidal output unit activation functions, takes the form of:

) = σ

(The *Source for the above neural network algorithm description is: DS-640: Neural Networks and Support Vector Machines - Robert Finn – Dec. 1, 2016)*

**Method**

For implementation, R programming packages LM and NEURAL NETWORK are used. The NEURAL NETWORK package is based on the back-propagation gradient descent algorithm. I am utilizing a double-hidden layer neural network for training.

I first used an R for loop to compute the returns from price using the formula: Returns = P(i)/P(i-1). I added it to the data set as separate column. After selecting the twenty indicators along with two columns: ticker and calendar date, I then used another R for loop to compute the log of returns and added it to the data set as a separate column. Because there are some zeros in the Returns column, using the log of returns rendered some R infinite (Inf) values in the Log of returns column. I converted those infinite values into “NAs”, before cleaning the data set from all the “NAs”.

I finally normalized all the factors using the following formula:

Normalized factor = (factor – mean (factor))/sd (factor), where mean is the mean of the column and sd is the standard deviation of the column.

I created a new R data frame consisting of all the twenty normalized factors, along with the calendar date column, the ticker column, the returns column and the log of returns column.

I found out the number of date in the data. And using an R for loop, I extracted all the data corresponding to a particular calendar date and created a data frame for each calendar date.

I will write R code to generate time series for the 20 betas derived from my linear regression of log-returns on your 20 chosen factors over the first three quarters of the dates for which I have data. I will then use these time series to forecast the 20 betas for the first date not included in the model which will give me an expected return for each stock in my model over the next time period. This expected return will then be employed to rank my stocks and split them into 5 groups with group 1 having the highest expected return and group 5 the lowest expected returns. The average actual returns for each group will then be calculated for the first date not included in the model. I will then continue this process on a rolling basis for the rest of the dates for which I have data. I will perform the same task using a neural network instead of a linear model.

**Results**

1. For each of the dates not included in the model, meaning the fourth quarter of each the years 2011 to 2015, I will predict the return for each stock with the neural network model, and use that to rank my stocks in a decreasing order before splitting them into 5 groups. I will then calculate the average returns for each group, and discuss the difference in returns for each group.
2. For the fourth quarter of 2011.

|  |  |  |
| --- | --- | --- |
| Groups | Average returns for 2011-12-31 | Predicted returns for 2011-12-31 |
| Group 1 | 0.11920804902 | 0.10329895143 |
| Group 2 | 0.11864927415 | 0.08354768423 |
| Group 3 | 0.10646070662 | 0.07487498111 |
| Group 4 | 0.05638041033 | 0.05834756905 |
| Group 5 | -0.02173637167 | -0.00277221546 |

**Table 1.1: Compared returns for 2011-12-31**

1. For the fourth quarter of 2012.

|  |  |  |
| --- | --- | --- |
| Groups | Average returns for 2012-12-31 | Predicted returns for 2012-12-31 |
| Group 1 | 0.114665701169 | 0.11928866714 |
| Group 2 | 0.115336406124 | 0.10229519698 |
| Group 3 | 0.127731203722 | 0.09334380542 |
| Group 4 | 0.093139804160 | 0.08427562973 |
| Group 5 | -0.008158161041 | 0.03373893868 |

**Table 1.2: Compared returns for 2012-12-31**

1. For the fourth quarter of 2013.

|  |  |  |
| --- | --- | --- |
| Groups | Average returns for 2013-12-31 | Predicted returns for 2013-12-31 |
| Group 1 | 0.12644519447 | 0.15312263061 |
| Group 2 | 0.06797968658 | 0.09074951366 |
| Group 3 | 0.05911131016 | 0.08574644381 |
| Group 4 | 0.05774970242 | 0.07914463761 |
| Group 5 | 0.04230322318 | 0.03131959371 |

**Table 1.3: Compared returns for 2013-12-31**

1. For the fourth quarter of 2014.

|  |  |  |
| --- | --- | --- |
| Groups | Average returns for 2014-12-31 | Predicted returns for 2014-12-31 |
| Group 1 | 0.108462193102 | 0.054992042596 |
| Group 2 | 0.048025840405 | 0.015478730918 |
| Group 3 | 0.029199241871 | 0.003751441872 |
| Group 4 | -0.009906632518 | -0.009045513456 |
| Group 5 | -0.159289343516 | -0.161057832384 |

**Table 1.4: Compared returns for 2014-12-31**

1. For the fourth quarter of 2015.

|  |  |  |
| --- | --- | --- |
| Groups | Average returns for 2015-12-31 | Predicted returns for 2015-12-31 |
| Group 1 | -0.11170813107 | -0.1566479089 |
| Group 2 | -0.09434602171 | -0.1827566447 |
| Group 3 | -0.12641122779 | -0.1982944464 |
| Group 4 | -0.16427263970 | -0.2162889483 |
| Group 5 | -0.33510869252 | -0.2934782144 |

**Table 1.5: Compared returns for 2015-12-31**

1. For each of the dates not included in the model, for the consumer services sector, I will predict the return for each stock in the neural network model, and use that to rank my stocks in a decreasing order before splitting them into 5 groups. I will then calculate the average returns for each group, and discuss the difference in returns for each group. My dates that are omitted from the model are the fourth quarter of each year, from 2011 to 2015.
2. For the fourth quarter of 2011.

|  |  |  |
| --- | --- | --- |
| Groups | Average returns for the consumer services sector for 2011-12-31 | Predicted returns for the consumer services sector for 2011-12-31 |
| Group 1 | 0.07440042386 | 0.19869675685 |
| Group 2 | 0.10258814887 | 0.10096677348 |
| Group 3 | 0.11422949794 | 0.07340414277 |
| Group 4 | 0.14370097771 | 0.03797191082 |
| Group 5 | 0.11073043774 | -0.08251064998 |

**Table 2.1: Compared returns for the consumer services sector for 2011-12-31**

1. For the fourth quarter of 2012.

|  |  |  |
| --- | --- | --- |
| Groups | Average returns for the consumer services sector for 2012-12-31 | Predicted returns for the consumer services sector for 2012-12-31 |
| Group 1 | 0.07292894974 | 0.18933516259 |
| Group 2 | 0.11057367223 | 0.12152389766 |
| Group 3 | 0.10853561096 | 0.10101327971 |
| Group 4 | 0.10615710824 | 0.07756437083 |
| Group 5 | 0.10503236729 | -0.04601719844 |

**Table 2.2: Compared returns for the consumer services sector for 2012-12-31**

1. For the fourth quarter of 2013

|  |  |  |
| --- | --- | --- |
| Groups | Average returns for the consumer services sector for 2013-12-31 | Predicted returns for the consumer services sector for 2013-12-31 |
| Group 1 | 0.08851569811 | 0.20063639357 |
| Group 2 | 0.00579667141 | 0.10890200423 |
| Group 3 | 0.04657280012 | 0.08450548208 |
| Group 4 | 0.03267138793 | 0.06234554423 |
| Group 5 | 0.03381618700 | 0.01236491519 |

**Table 2.3: Compared returns for the consumer services sector for 2013-12-31**

1. For the fourth quarter of 2014.

|  |  |  |
| --- | --- | --- |
| Groups | Average returns for the consumer services sector for 2014-12-31 | Predicted returns for the consumer services sector for 2014-12-31 |
| Group 1 | 0.02090692256 | 0.09338150278 |
| Group 2 | 0.08256094767 | 0.03463564371 |
| Group 3 | 0.06995810172 | 0.01062666626 |
| Group 4 | 0.05663439627 | -0.01762880476 |
| Group 5 | 0.01174023502 | -0.05663913439 |

**Table 2.4: Compared returns for the consumer services sector for 2014-12-31**

1. For the fourth quarter of 2015.

|  |  |  |
| --- | --- | --- |
| Groups | Average returns for the consumer services sector for 2015-12-31 | Predicted returns for the consumer services sector for 2015-12-31 |
| Group 1 | -0.13872101045 | -0.09485631747 |
| Group 2 | -0.08766135935 | -0.18121498818 |
| Group 3 | -0.05554749332 | -0.20111631842 |
| Group 4 | -0.12320794394 | -0.23885325848 |
| Group 5 | -0.17460689546 | -0.39248021351 |

**Table 2.5: Compared returns for the consumer services sector for 2015-12-31**

**Discussion**

1. **Performance of the model**
2. For the fourth quarter of 2011, a comparison between the average log of returns and the predicted returns using neural network shows that they are lower and pretty close for group 1 where the average return is 0.12 and the predicted return is 0.10. For group 2, predictions are little lower, with an average return of 0.12 and a predicted return of 0.08. We see the same trend for group 3 with an average return of 0.11 and a predicted return of 0.07. Predictions are close and a little higher for group 4 with an average return of 0.056 and a predicted return of 0.058. Predictions for group 5 are higher with an average return of -0.0217 and a predicted return of -0.0028. Returns for group 5 remain in negative territory for this quarter.
3. For the fourth quarter of 2012, a comparison between the average log of returns and the predicted returns using neural network shows that overall they are pretty close for all the groups. Predictions are higher for group 1 with an average return of 0.11 and a predicted return of 0.12. For group 2, predictions are little lower, with an average return of 0.12 and a predicted return of 0.10. We see the same trend for group 3 with an average return of 0.13 and a predicted return of 0.09. Predictions are lower but very close for group 4 with an average return of 0.09 and a predicted return of 0.08. Predictions are higher for group 5, with an average return of -0.008 and a predicted return of 0.034.
4. The fourth quarter of 2013 has almost similar trends from the fourth quarter of 2012. A comparison between the average log of returns and the predicted returns using neural network shows that overall they are pretty close for all the groups. Predictions are higher for group 1 with an average return of 0.13 and a predicted return of 0.15. For group 2, predictions are little higher and very close, with an average return of 0.07 and a predicted return of 0.09. Same trend for group 3, Predictions are higher and very close, with an average return of 0.06 and a predicted return of 0.09. Predictions are higher and very close for group 4 with an average return of 0.06 and a predicted return of 0.08. Predictions are lower for group 5, but remain very close, with an average return of 0.04 and predicted returns of 0.03.
5. For the fourth quarter of 2014, a comparison between the average log of returns and the predicted returns using neural network shows that overall they are pretty close for all the groups. Predictions are little lower for group 1 with an average return of 0.11 and a predicted return of 0.05. For group 2, predictions are also lower, with an average return of 0.05 and a predicted return of 0.02. We see the same trend for group 3 with an average return of 0.03 and a predicted return of 0.004. Predictions are higher and very close for group 4 with an average return of -0.0099 and a predicted return of -0.0090. Predictions are lower for group 5, with an average return of -0.159 and a predicted return of -0.161. Predictions for groups 4 and 5 remain in negative territory.
6. For the fourth quarter of 2015, a comparison between the average log of returns and the predicted returns using neural network shows that overall they are lower and pretty close for all the groups. Predictions are little lower for group 1 with an average return of -0.11 and a predicted return of -0.16. For group 2, predictions are also little lower, with an average return of -0.09 and a predicted return of -0.18. We see the same trend for group 3 with an average return of -0.126 and a predicted return of -0.198. Predictions are lower and very close for group 4 with an average return of -0.16 and a predicted return of -0.22. Predictions are higher for group 5, and remain very close, with an average return of -0.34 and a predicted return of -0.29.
7. **Applying the model to the Consumer Services Sector**
8. For the fourth quarter of 2011, a comparison between the average log of returns and the predicted returns using neural network shows that overall they are pretty close for all the groups. Predictions are higher for group 1 with an average return of 0.07 and a predicted return of 0.20. For group 2, predictions are little lower, with an average return of 0.1026 and a predicted return of 0.1010. Predictions are lower for group 3 with an average return of 0.11 and a predicted return of 0.07. Predictions are lower for group 4 with an average return of 0.14 and a predicted return of 0.04. We see a similar trend for group 5, lower and close, with an average return of 0.11 and a predicted return of -0.08. Predictions for group 5 reached negative territory for this quarter.
9. For the fourth quarter of 2012, a comparison between the average log of returns and the predicted returns using neural network shows that overall they are pretty close for all the groups. Predictions are little higher for group 1 with an average return of 0.07 and a predicted return of 0.19. For group 2, predictions are little higher, with an average return of 0.11 and a predicted return of 0.12. Predictions are lower but very close for group 3 with an average return of 0.11 and a predicted return of 0.10. We have similar trend for group 4 with an average return of 0.11 and a predicted return of 0.08. Predictions for group 5 are lower, with an average return of 0.11 and a predicted return of -0.05. Predictions for group 5 went from positive to negative for this quarter.
10. For the fourth quarter of 2013, a comparison between the average log of returns and the predicted returns using neural network shows that overall they are pretty close for all the groups. Predictions are little higher for group 1 with an average return of 0.09 and a predicted return of 0.20. For group 2, predictions are little higher, with an average return of 0.006 and a predicted return of 0.11. We have the same trend for group 3 with an average return of 0.05 and a predicted return of 0.08. Predictions are higher and close for group 4 with an average return of 0.03 and a predicted return of 0.06. For group 5, predictions are lower with an average return of 0.03 and a predicted return of 0.01.
11. For the fourth quarter of 2014, a comparison between the average log of returns and the predicted returns using neural network shows that overall they are pretty close for all the groups. Predictions are little higher for group 1 with an average return of 0.02 and a predicted return of 0.10. For group 2, predictions are lower, with an average return of 0.08 and a predicted return of 0.03. We have the same trend for group 3 with an average return of 0.07 and a predicted return of 0.01. Predictions are lower and for group 4 with an average return of 0.06 and a predicted return of -0.02. We have similar trend for group 5 with an average return of 0.01 and a predicted return of -0.06. Predictions for groups 4 and 5 went from positive to negative for this quarter.
12. For the fourth quarter of 2015, a comparison between the average log of returns and the predicted returns using neural network shows that overall they are pretty close for all the groups. Predictions are little higher for group 1 with an average return of -0.14 and a predicted return of -0.09. For group 2, predictions are little lower, with an average return of -0.09 and a predicted return of -0.18. We have same trend for group 3 with an average return of -0.06 and a predicted return of -0.2. Predictions are also lower and for group 4 with an average return of -0.12 and a predicted return of -0.24. Similar trend for group 5, lower, with an average return of -0.17 and a predicted return of -0.39.

**Conclusion**

Overall, the double-hidden layer Neural Network model has performed better than Linear Regression model. Predicted returns are closer to the actual average returns using neural network. The only limitation is that it takes time to train the neural network, which leads to a slow convergence. If we increase the threshold during the training of the neural network, it will converge faster but the precision of the predictions will be affected. Many more trainings of the neural network should be conducted in order for the model to improve.

**References**

1. *A.K. Parida, R. Bisoi, P.K. Dash, S. Mishra. International Journal of Computational Intelligence Systems, Vol. 10 (2017) 375–393*
2. *Chakraborty, K., Mehrotra, K., Mohan, C.K., Ranka, S., 1992. Forecasting the behavior of multivariate time series using neural networks. Neural Netw. 5, 961–970. http:// dx.doi.org/10.1016/S0893-6080(05)80092-9*
3. *Croston, J.D., 1972. Forecasting and stock control for intermittent demands. J. Oper. Res. Soc..* [*http://dx.doi.org/10.1057/jors.1972.50*](http://dx.doi.org/10.1057/jors.1972.50)*.*
4. *Deyun Wanga, Hongyuan Luoa, Olivier Grunderc, Yanbing Lina, Haixiang Guoa. Applied Energy Volume 190, 15 March 2017, Pages 390–407*
5. *F. Lollia, R. Gamberinia, A. Regattierib, E. Balugania, T. Gatosb, S. Guccib. International Journal of Production Economics. Volume 183, Part A, January 2017, Pages 116–128*
6. *Gutierrez, R.S., Solis, A.O., Mukhopadhyay, S., 2008. Lumpy demand forecasting using neural networks. Int. J. Prod. Econ. 111, 409–420. http://dx.doi.org/10.1016/ j.ijpe.2007.01.007.*
7. *Hill, T., O’Connor, M., Remus, W., 1996. Neural network models for time series forecasts. Manag. Sci. 42, 1082–1092.* [*http://dx.doi.org/10.1287/mnsc.42.7.1082*](http://dx.doi.org/10.1287/mnsc.42.7.1082)*.*
8. *Huang, G.-B., Zhu, Q., Siew, C., Ã, G.H., Zhu, Q., Siew, C., Huang, G.-B., Zhu, Q., Siew, C., 2006. Extreme learning machine: theory and applications. Neurocomputing 70, 489–501.* [*http://dx.doi.org/10.1016/j.neucom.2005.12.126*](http://dx.doi.org/10.1016/j.neucom.2005.12.126)*.*
9. *Huang, G., Huang, G.-B., Song, S., You, K., 2015. Trends in extreme learning machines: a review. Neural Netw. 61, 32–48.* [*http://dx.doi.org/10.1016/j.neunet.2014.10.001*](http://dx.doi.org/10.1016/j.neunet.2014.10.001)*.*
10. *Huang, G.-B., Babri, H.A., 1998. Upper bounds on the number of hidden neurons in feedforward networks with arbitrary bounded nonlinear activation functions. IEEE Trans. Neural Netw. 9, 224–229.* [*http://dx.doi.org/10.1109/72.655045*](http://dx.doi.org/10.1109/72.655045)*.*
11. *Huang, G.-B., 2003. Learning capability and storage capacity of two-hidden-layer feedforward networks. IEEE Trans. Neural Netw. 14, 274–281. http://dx.doi.org/ 10.1109/TNN.2003.809401.*
12. *Inani K. S., Tripathi M., Kumar S. The Journal of Prediction Markets, Vol.10, No 2 (2016)*
13. *K. Ashwin Kumar, T. Niranjan Babu, Nitish Vaishy, K. Lavanya. International Journal of Advanced Research in Computer Science, Vol. 7, No 7 (2016)*
14. *Kourentzes, N., 2013. Intermittent demand forecasts with neural networks. Int. J. Prod. Econ. 143, 198–206.* [*http://dx.doi.org/10.1016/j.ijpe.2013.01.009*](http://dx.doi.org/10.1016/j.ijpe.2013.01.009)*.*
15. *Kuang, Y., Singh, R., Singh, S. et al. Multimedia Tools and Application (2017). doi:10.1007/s11042-016-4319-9*
16. *Mukhopadhyay, S., Solis, A.O., Gutierrez, R.S., 2012. The accuracy of non-traditional versus traditional methods of forecasting lumpy demand. J. Forecast. 31, 721–735.* [*http://dx.doi.org/10.1002/for.1242*](http://dx.doi.org/10.1002/for.1242)*.*
17. *Patra, A., Das, S., Mishra, S.N. et al. Neural Comput & Applic (2017) 28: 101. doi:10.1007/s00521-015-2039-0*
18. *Prof. Finn R., Saint Peter’s University’s Data Science Program, Course DS-640-Predictive Analytics and Expert Design: Neural Networks and Support Vector Machines, Dec. 1, 2016*
19. *Satapathy P., Dhar S., "A hybrid functional link extreme learning machine for Maximum Power Point Tracking of partially shaded Photovoltaic array", Power Communication and Information Technology Conference (PCITC) 2015 IEEE, pp. 409-416, 2015.*
20. *Syntetos, A.A., Boylan, J.E., Croston, J.D., 2005. On the categorization of demand patterns. J. Oper. Res. Soc. 56, 495–503* [*http://www.jstor.org/stable/41021*](http://www.jstor.org/stable/41021)*.*