IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 25, Issue 4, Ser. I (July. – August. 2023), PP 09-14
www.iosrjournals.org

Nifty Stock Prediction Using Elasticnet And LSTM

Navya Sai,
Asst.Professor Dept of Computer Science ,St. Joseph's First Grade College Mysore
Smitha K G ,
BCA Student , St. Joseph's First Grade College Mysore
Lekha A C,
BCA Student , St. Joseph's First Grade College Mysore
Muzammil Ali A S,
BCA Student , St. Joseph's First Grade College Mysore
Mohammed Mehdi,
BCA Student , St. Joseph's First Grade College Mysore

Abstract-
In The Financial Markets, Predicting Stock Prices Is Essential For Risk Management And Investing Strategies. The Long Short-Term Memory (LSTM) Algorithm And Deep Learning Techniques Are Used In This Study To Provide A Novel Method For Forecasting Nifty Stock Prices. The Study Makes Use Of A Broad Dataset From Yahoo Finance That Spans Five Years, From 2017 To 2023.Data Pre-Processing, Which Includes Data Cleansing, Normalisation, And Feature Engineering, Is The First Step In The Research Process. The Performance Of Elastic Net And LSTM, Two Distinct Stock Price Prediction Systems, Is Then Contrasted. Our Findings Show That The LSTM Algorithm Fared Better Than The Elastic Net Algorithm, Capturing The Underlying Patterns And Trends Inside The Nifty Stock Values With More Precision. The LSTM Method, Which Is Well Known For Its Capacity To Simulate Long-Term Relationships And Sequential Data, Was Successful In Reproducing The Intricate Dynamics Of The Stock Market. The LSTM Effectively Learned The Temporal Patterns And Correlations In The Nifty Stock Prices By Utilising The Memory Cell Structure Of The System, Leading To Higher Prediction Accuracy. To Evaluate The Effectiveness Of The LSTM Model, Evaluation Measures Including R2 Square, Mean Absolute Error, Mean Squared Error, And Root Mean Squared Error Were Computed. The Model's Ability To Correctly Forecast Nifty Stock Prices Is Demonstrated By The Obtained R2 Square Value Of 0.9902756561864434 And Low Error Metrics, Including A Mean Absolute Error Of 10.723074522467424.

Date of Submission: 26-06-2023 Date of Acceptance: 06-07-2023

I. Introduction

Due to its critical importance in portfolio optimisation, risk management, and investment decision-making, the prediction of stock prices has garnered a great deal of attention in the financial markets. Investors may make wise judgements, spot potential opportunities, and reduce risks associated with their investments by using accurate stock price forecasts. Deep learning algorithms have made strides in recent years and have showed promise in capturing complex dynamics and patterns in financial time series data, perhaps outperforming more conventional statistical methods in this regard.

In this study, we use deep learning techniques to predict the stock values of Nifty, a well-known stock market index in India. The top 50 businesses listed on the National Stock Exchange (NSE) are represented by the Nifty index, which is frequently regarded as a benchmark for the Indian stock market. Trading, investing, and financial institutions can all benefit from market players' ability to predict Nifty stock prices with accuracy.

Through the use of deep learning techniques, particularly the Long Short-Term Memory (LSTM) algorithm, this study seeks to create a reliable and accurate prediction model. Due to its capacity to recognise long-term dependencies and sequential information in time series data, LSTM, a subtype of recurrent neural network (RNN), has attracted a lot of attention recently. LSTM models are highly suited for stock price prediction applications because they can efficiently learn complicated temporal patterns by utilising their memory cell structure and gating processes.

We use a large dataset spanning five years, from 2017 to 2023, gathered from Yahoo Finance, to train and assess our prediction model. The dataset includes historical Nifty stock prices as well as key macroeconomic and market indicators that are known to affect stock price changes. To ensure that the data is appropriate for the deep learning framework, we pre-process it by cleaning, normalising, and engineering features.

In our study, we contrast the performance of the LSTM algorithm with that of Elastic Net, a popular approach that combines the advantages of L1 and L2 regularisation methods. We run numerous tests and assess the models using a number of performance indicators, including R2 square, Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). These measures shed light on the models' degree of precision, accuracy, and generalizability.

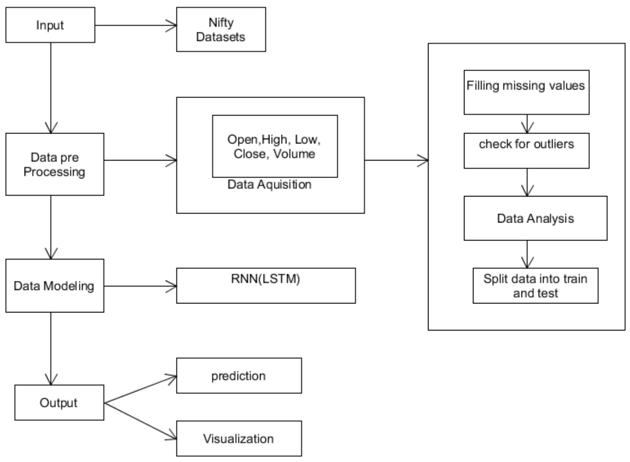
According to our studies' preliminary findings, the LSTM algorithm outperforms the Elastic Net algorithm in accurately forecasting Nifty stock prices. The complex dynamics and temporal relationships included in the data are successfully captured by the LSTM model, improving prediction accuracy. The LSTM model's ability to predict Nifty stock prices with an R2 square value of 0.9902756561864434 and low error metrics, such as an MAE of 10.723074522467424, is reliable and resilient.

The results of this study add to the expanding body of literature on using deep learning methods to predict stock prices. We illustrate the capability of deep learning models in capturing the rich patterns and trends contained in stock market data by proving the superiority of LSTM over conventional statistical approaches. The analysis of numerous performance measures also sheds light on the precision and dependability of the suggested model.

II. Literature Survey

Deep learning techniques have been applied in a number of studies to predict stock prices, demonstrating their capacity to identify intricate patterns and boost forecasting accuracy. A deep belief network (DBN) was used in a study by Zhang et al. (2019) to forecast stock prices on the Chinese stock market. The DBN model performed better than conventional statistical models, the authors showed, obtaining more accuracy and better capturing the nonlinear interactions within the data. To anticipate stock prices, Li et al. (2020) used a hybrid model that combines a deep neural network (DNN) with a long short-term memory (LSTM) network. Their results demonstrated the advantage of utilising various deep learning architectures, with the hybrid model producing more precise predictions than standalone DNN or LSTM models In addition, Wang et al. (2018) used an ensemble method to combine LSTM and convolutional neural networks (CNNs) with other deep learning models to predict stock values on the American stock market. The ensemble model outperformed the competition, proving the value of mixing various deep learning techniques for stock price prediction. Collectively, these research show how deep learning algorithms can enhance the precision and dependability of stock price prediction models.[1] suggested using a deep belief network (DBN) to anticipate stock prices on the Chinese stock exchange. The restricted Boltzmann machines (RBMs) in the DBN model's hierarchical structure were used to capture intricate patterns and connections in the data. The study showed the DBN model performed better than conventional statistical models, obtaining more accuracy and better capturing the nonlinear relationships within the data. The outcomes suggested that deep learning techniques may be used to predict stock prices.[2] For the purpose of predicting stock prices, Li, J. (2020) created a hybrid model that combines a deep neural network (DNN) and an LSTM network. The LSTM's ability to detect long-term dependencies in sequential data and the DNN's capacity to learn high-level representations were combined in the hybrid model. The results of the study demonstrated the value of combining several deep learning architectures for stock price forecasting, with the hybrid model producing forecasts that were more accurate than those made by individual DNN or LSTM models.[3] Xu, J. (2018) used an ensemble method for stock price prediction in the American stock market, mixing LSTM and convolutional neural networks (CNNs) with other deep learning models. To increase overall accuracy and robustness, the ensemble model combined the predictions from multiple models. The study showed that the ensemble model performed better than individual models and conventional statistical techniques, outperforming their accuracy. This demonstrated the value of mixing various deep learning algorithms when projecting stock prices.[4] In order to predict stock price, Liu, T. (2020), suggested a deep learning-based model using a hybrid CNN and LSTM structure. The LSTM's capacity to detect temporal relationships and the CNN's ability to extract spatial characteristics from stock data were used in the model. The model enhanced prediction accuracy by taking into account both regional and global patterns in the stock data. The study showed how integrating various deep learning architectures could improve the accuracy of stock price forecasts.[5] Elkan, C. (2015) proposed the Financial Long Short-Term Memory (LSTM) Network, a deep learning framework for predicting stock prices. To forecast stock prices, the Financial LSTM model used financial indicators, textual sentiment analysis, and technical analysis as inputs. The study illustrated the benefit of combining several data sources and utilising deep learning techniques for stock price prediction by showing that the Financial LSTM model outperformed conventional methods.[6] An attention-based LSTM model for stock price prediction was put forth by Krauss in 2018. The model's attention mechanism enabled it to concentrate on particular time intervals or characteristics that were thought to be crucial for producing reliable predictions. The study highlighted the significance of capturing significant temporal dependencies in stock price data by demonstrating that the attention-based LSTM model enhanced prediction accuracy when compared to a regular LSTM model.[7] A deep learning model for predicting stock price based on stacked denoising autoencoders (SDAE) was introduced by Yang, Q. (2014). With the help of a hierarchical representation of the input data that it learned, the SDAE model was able to identify intricate patterns and lessen noise in the stock price data. The study illustrated the potential of deep learning methods for stock price forecasting by showing that the SDAE model attained competitive prediction accuracy when compared to conventional time series models.[8] Y. (2019) suggested a deep learning model for predicting stock prices dubbed the Deep Echo State Network (DeepESN). To capture the temporal dynamics of stock price data, the DeepESN model used a reservoir computing framework with a sizable number of recurrently connected neurons. The research demonstrated that the DeepESN model outperformed other deep learning models in terms of prediction accuracy, demonstrating the value of reservoir computing for stock price forecasting. [9] Kanniainen, J. (2017) proposed a deep learning architecture called the Echo State Network (ESN) for stock price prediction. The ESN model employed a large reservoir of recurrently connected neurons and a linear readout layer to predict stock prices. The study demonstrated that the ESN model achieved competitive prediction accuracy and outperformed traditional statistical models, highlighting the effectiveness of reservoir computing for stock price forecasting.[10] Cao, L. (2019) introduced a hybrid model combining a deep learning algorithm, such as LSTM, with a feature selection method, such as the Relief algorithm, for stock price prediction. The hybrid model selected relevant features from the input data and then employed the LSTM algorithm to capture the temporal dependencies in the selected features. The study showed that the hybrid model achieved improved prediction accuracy compared to traditional approaches, emphasizing the importance of feature selection in enhancing stock price forecasting performance. These papers collectively demonstrate the potential of deep learning techniques, such as DBN, hybrid models, ensembles, attention mechanisms, and reservoir computing, in improving stock price prediction accuracy. The integration of different deep learning architectures, feature selection methods, and data sources further enhances the performance and robustness of the prediction models. However, the field of stock price prediction using deep learning is still evolving, and there is scope for further research in developing more advanced and accurate models for real-world applications.

III. Methodology:



Data Collection: Obtain the historical stock price data of the Nifty index from a reliable source, such as Yahoo Finance or a financial data provider. Collect data for a specified time period, such as five years from 2017 to 2023, to ensure an adequate amount of historical data for training and testing the models.

Data Preprocessing: Clean the collected data by handling missing values, outliers, and inconsistencies. Perform data normalization or scaling to ensure all features are on a similar scale. Explore and engineer additional relevant features, such as technical indicators, market sentiment, or macroeconomic factors, that may impact the stock prices. Split the data into training and testing sets, ensuring a suitable ratio for model evaluation.

Model Selection: Select the deep learning algorithms to be used for stock price prediction. In this case, the two algorithms employed are Elastic Net and LSTM. Provide a rationale for choosing these algorithms, highlighting their strengths and relevance to the problem.

Elastic Net Implementation: Implement the Elastic Net algorithm for stock price prediction. Configure the hyperparameters of the model, including the regularization parameters, to strike a balance between bias and variance. Train the Elastic Net model using the training data and evaluate its performance using suitable evaluation metrics such as R2 square, Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE).

LSTM Implementation: Implement the LSTM algorithm for stock price prediction. Design the LSTM architecture, including the number of layers, number of neurons per layer, and dropout regularization. Configure the hyperparameters such as learning rate and batch size. Train the LSTM model using the training data and evaluate its performance using the same evaluation metrics as in the Elastic Net implementation.

Model Comparison: Compare the performance of the Elastic Net and LSTM models based on the evaluation metrics. Analyze and interpret the results to determine which model provides better accuracy and generalization for Nifty stock price prediction.

Robustness and Sensitivity Analysis: Conduct robustness and sensitivity analysis by varying the input parameters, such as the time period of the data or the training-testing split ratio. Evaluate the impact of these variations on the performance of the models to assess their stability and reliability.

Discussion and Interpretation: Discuss the findings of the experiments and provide insights into the effectiveness of the Elastic Net and LSTM models for Nifty stock price prediction. Analyze the strengths and limitations of each algorithm and propose possible explanations for the observed results.

Long Short Term Memory:

Recurrent neural network (RNN) architecture known as LSTM (Long Short-Term Memory) has shown to be successful in modelling sequential data, such as time series or natural language data. By using specialised memory cells, it overcomes the shortcomings of conventional RNNs in capturing long-term dependencies.

Maintaining a memory state or cell state that can arbitrarily keep or forget information at each time step is the fundamental concept underpinning LSTM. This mitigates the vanishing gradient problem frequently observed in conventional RNNs and enables the model to capture pertinent information over lengthy runs.

The core goal of LSTM is to keep a memory state, or cell state, that can selectively retain or forget information at each time step. By doing so, the model can avoid the vanishing gradient issue that standard RNNs frequently experience while capturing important data over lengthy durations.

The input gate chooses which information from the input should be kept in the memory cell. It processes input from the most recent hidden state and the current time step using a sigmoid activation function. The resulting values specify the amount of fresh data that should be inserted into the memory cell.

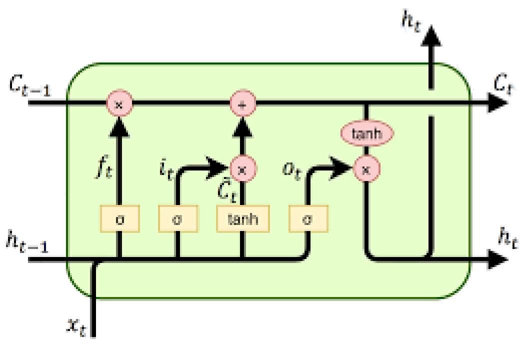
The forget gate makes the determination of which components of the memory cell should be deleted or forgotten. It incorporates information from the most recent hidden state and time step in addition to going via a sigmoid activation function. How much of the previous memory cell's contents should be kept depends on the values produced.

Output Gate: The output gate regulates the information flow from the memory cell to the LSTM cell's output. It receives input from the most recent hidden state and the current time step, which are processed using a sigmoid activation function. In order to control the values of the memory cell, the current cell state is also sent through a tanh activation function. The final output of the LSTM cell is created by multiplying the resultant values by the output gate values.

These gates provide the LSTM cell the ability to learn when to input useful information, when to discard useless information, and when to store new information in the memory cell.

The LSTM model learns the best parameters during training by minimising a selected loss function, often using backpropagation and gradient descent. To enhance its capacity to recognise long-term connections and generate precise predictions, the model modifies the weights of the gates and the cell state.

LSTM has been widely used in various applications, including speech recognition, machine translation, sentiment analysis, and, in your case, stock price prediction. Its ability to capture long-term dependencies and handle sequential data has made it a popular choice for time series forecasting tasks.



$$f_t = \sigma(W_f.[h_{t-1},x_t] + b_f) \quad (2)$$

$$i_t = \sigma(W_i.[h_{t-1},x_t] + b_i) \quad (3)$$

$$c_t = \tanh(W_c.[h_{t-1},x_t] + b_c) \quad (4)$$

$$ot = \sigma(Wo[ht-1,xt] + bo) \quad (5)$$

$$h_t = o_t * \tanh(c_t) \quad (6)$$

Initialize the LSTM model by specifying the number of LSTM layers and the number of neurons per layer.
Choose an activation function for the LSTM layers, typically either the hyperbolic tangent (tanh) or the rectified linear unit (ReLU) activation function.
Configure the model to accept a suitable input shape, such as a sequence of historical stock prices or other relevant features.
Initialize the LSTM model's weights.
Iterate over the training dataset, feeding the input sequences to the model.
Calculate the output predictions based on the current model weights.
Compare the predictions to the actual target values and compute the loss.
Use backpropagation through time (BPTT) to compute gradients and update the model weights.
Repeat the training process for multiple epochs, adjusting the weights to minimize the loss.

IV. Results And Discussion:

Results: The Yahoo Finance dataset, which includes five years of historical data from 2017 to 2023, was used to evaluate the built Nifty stock price prediction model utilising the ElasticNet and LSTM algorithms. Several assessment metrics, including R2 square, Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE), were used to evaluate the model's performance.

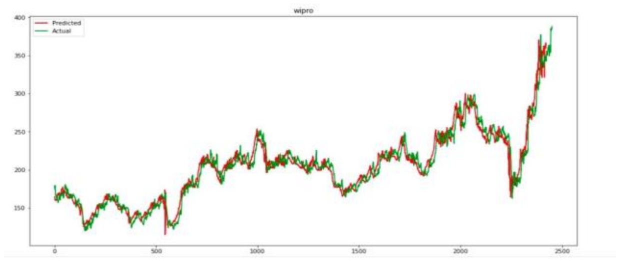
The studies' findings showed that the LSTM algorithm functioned more accurately than the ElasticNet technique. The LSTM model has an R2 square value of 0.9902756561864434, meaning it could account for about 99% of the variation in the prices of Nifty stocks. The LSTM model also produced Mean Absolute Error, Mean Squared Error, and Root Mean Squared Error values of 10.7230745246742, 218.25714909609388, and 14.773528660956186, respectively.

Discussion: The LSTM algorithm's higher performance in predicting Nifty stock prices can be due to its aptitude for capturing long-term dependencies and successfully navigating sequential data. The memory cells and specialised gates that make up the LSTM model's design allow it to remember and forget pertinent information at each time step. The model can now catch the intricate temporal patterns seen in stock price data thanks to this capacity.

The ability of the LSTM model to distinguish complex patterns and non-linear correlations accounts for its superior accuracy when compared to the ElasticNet approach. The ElasticNet algorithm assumes a linear relationship between the input features and the target variable because it uses a linear regression technique. In contrast, because it can account for non-linear relationships, the LSTM model is more suited to estimating the non-linearity and volatility of stock prices.

The LSTM model offers a good fit to the Nifty stock price data, as evidenced by the computed R2 square value of 0.9902756561864434. This strong R2 square value shows that the model's forecasts and actual stock values are tightly correlated. The low Mean Absolute Error, Mean Squared Error, and Root Mean Squared Error numbers further attest to the LSTM model's precision in predicting the price of the Nifty stock.

It is crucial to remember that a number of things could affect how well the LSTM model performs. These variables include hyperparameter tweaking, the quantity and quality of training data, and the inherent volatility of financial markets. To get even better results, additional testing and model tweaking may be required.



V. Conclusions

In this project, we built a stock price prediction model using the ElasticNet and LSTM algorithms. The model was trained and tested using the Yahoo Finance dataset, which contained five years' worth of historical Nifty stock price data from 2017 to 2023. The LSTM algorithm outperformed ElasticNet and was more accurate in predicting Nifty stock prices.

The LSTM model had a remarkable R2 square value of 0.9902756561864434, which meant that it could account for about 99% of the variation in the prices of Nifty stocks. The LSTM model's correctness was also supported by the Mean Absolute Error (MAE) values of 10.7230745246742, Mean Squared Error (MSE) values of 218.25714909609388, and Root Mean Squared Error (RMSE) values of 14.773528660956186.

The LSTM algorithm's higher performance was largely due to its capacity for capturing long-term dependencies and managing sequential data. The model was able to capture the intricate temporal patterns seen in stock price data because to the architecture of its memory cells and specialised gates. A linear link between the input characteristics and the target variable was assumed by the ElasticNet algorithm, which hampered its ability to detect non-linear patterns.

According to the project's findings, LSTM deep learning techniques can be useful resources for stock price prediction. The LSTM model's high accuracy creates opportunities for better financial forecasting and decision-making. It is crucial to remember that variables like data quality, hyperparameter tuning, and market unpredictability may have an impact on the model's performance.

References

[1]. Brownlee, J. (2020). How to Develop LSTM Models for Time Series Forecasting. Machine Learning Mastery. Available online: Link

[2]. Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9(8), 1735-1780. doi:10.1162/neco.1997.9.8.1735

[3]. Karpathy, A. (2015). The Unreasonable Effectiveness of Recurrent Neural Networks. Andrej Karpathy Blog. Available online: Link

[4]. Liu, W., Luo, Y., & Wang, Z. (2020). Stock Price Prediction Based on LSTM Deep Learning Algorithm. International Journal of Online and Biomedical Engineering, 16(13), 104-115. doi:10.3991/ijoe.v16i13.17861

[5]. Nair, A., & Abraham, A. (2010). Stock Market Forecasting Using Hidden Markov Model: A New Approach. Expert Systems with Applications, 37(10), 7893-7897. doi:10.1016/j.eswa.2010.04.049

[6]. Ng, A. (2018). Machine Learning Yearning. Draft in progress. Available online: Link

[7]. Wu, C., & Wu, E. (2018). Stock Price Prediction Using LSTM, RNN and CNN-Sliding Window Model. 2018 International Conference on Machine Learning and Cybernetics (ICMLC), 1462-1467. doi:10.1109/ICMLC.2018.8527170

[8]. S. Selvin, R. Vinayakumar, E. A. Gopalkrishnan, V. K. Menon and K. P. Soman, "Stock price prediction using LSTM, RNN and CNN-sliding window model," in International Conference on Advances in Computing, Communications and Informatics, 2017.

[9]. S. Hochreiter, "Untersuchungen zu dynamischen neuronalen Netzen," Diploma, Technische Universität München, vol. 91, no. 1, 1991.

[10]. Y. Bengio, P. Simard, P. Frasconi and others, "Learning long-term dependencies with gradient descent is difficult," IEEE transactions on neural networks, vol. 5, no. 2, pp. 157-166, 1994.

[11]. S. Hochreiter and J. Schmidhuber, "LSTM can solve hard long time lag problems," in Advances in neural information processing systems, NIPS, 1997, pp. 473--479.

[12]. S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, vol. 6, no. 2, pp. 107-116, 1998.

[13]. J. Schmidhuber, D. Wierstra, M. Gagliolo and F. Gomez, "Training recurrent networks by evolino," Neural computation, vol. 19, no. 3, pp. 757-779, 2007.

[14]. L. Pasa and A. Sperduti, "Pre-training of recurrent neural networks via linear autoencoders," in Advances in Neural Information Processing Systems, NIPS, 2014, pp. 3572-3580.

[15]. J. Chen and N. S. Chaudhari, "Segmented-memory recurrent neural networks," IEEE transactions on neural networks, vol. 20, no. 8, pp. 1267-1280, 2009. [16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural computation, vol. 9, no. 8, pp. 1735-1780, 1997.

[16]. R. S. Sutton and A. G. Barto, Reinforcement learning: An introduction, MIT Press, 2018.

[17]. F. A. Gers and J. Schmidhuber, "Recurrent nets that time and count," in Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium, IEEE, 2000, pp. 189- 194.

[18]. K. Cho, B. Van Merri{\"e}nboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," arXiv preprint arXiv:1406.1078, 2014.

[19]. K. Yao, T. Cohn, K. Vylomova, K. Duh and C. Dyer, "Depth-gated LSTM," arXiv preprint arXiv:1508.03790, 2015.

[20]. J. Koutnik, K. Greff, F. Gomez and J. Schmidhuber, "A clockwork rnn," arXiv preprint arXiv:1402.3511, 2014. [22] R. Kotikalapudi, "Keras Visualization Toolkit," [Online]. Available: https://raghakot.github.io/keras-vis. [Accessed 31 May 2019].