# Technical Updates

I will be suggesting 4 major technical updates that can be taken up in consideration (none of them are more complex than the method proposed in the paper).

1. Update in Pre-processing of the Data
2. Improving the defects of MSE and time windowing
3. Update in the method used for RNN
4. A lighter and simpler model instead of RNN i.e. CNN

Apart from these updates, simpler updates like using a multivariate approach instead of a univariate approach and using ARIMA (SARIMA) for the threshold values have already been discussed in the critical analysis.

## 1. Update in Pre-processing of the Data

Introduction of the concept of stationarity of data during pre-processing and cross checking through Dicky Fuller Test can be a method that can drastically improve the model's performance. Stationarity is a property of time series data which implies that the distributional properties (mean/variance) have not changed over a period of time. For time series forecasting, stationarity is important because if stationary data is not present, we are practically expecting the model to predict what it has never seen before.

Inspired from: https://github.com/tklouie/PyData_LA_2018/blob/master/PyData_LA_2018_Tutorial.ipynb ,this repository explains how to make a dataset stationary.

## 2. Improving the defects of MSE and the method of time windowing

This update takes in account the increase in the number of epochs/iterations/trails during training, which helps us eradicate stagnancy in the MSE (Mean Squared Error). To ensure this, we take many iterations and take MSE of each step. We specify the iteration to get stopped when MSE hits a particular value (mostly it is the benchmark value determined by a single iteration or determined by ARIMA method). We make sure to break the loop as soon as MSE starts becoming constant or increasing. This ensures we have the minimum possible error function, removing the problem of stagnancy in the MSE. We can also use percentage errors instead of MSE, and follow the same analogy.

Also, using methods like rolling/sliding time windows instead of fixed time windowing can be very useful. (More on: https://machinelearningmastery.com/time-series-forecasting-supervised-learning)

## 3. Update in the method used in RNN

Recurrent Neural Networks are discussed vaguely in the paper, but the mechanism is still unknown. So, to provide a little transparency the method that should be used is LTSM (Long Short Term Memory Network). LTSMs build short and long term memories by revealing data to the hidden layers/nodes/units in a sequential fashion. LTSM uses the concept of working memory that is temporary which serves as an input for the regression layer. Out of different types of RNNs, LTSMs should be chosen because they can learn from long sequences while vanilla RNNs may not.

## 4. A lighter and simpler model instead of an RNN, i.e CNN

A Convolutional Neural Network can extract information from the temporal structure of the data by preserving the temporal structure of the time series in the input layer and using filters which can look for patterns to extract features. This gives us two advantages; we indirectly use a multivariate approach to train the model and CNN being a lighter model than RNN, works faster. The results are not compromised to a huge extent and in many cases, CNNs turn out to be performing better than LTSMs. The information extracted by the filters are known as a feature map and each additional feature results in a new feature map extracting a more complex feature. In time series, a filter can only move along 1 dimension i.e. time, hence the CNNs we use for time series prediction are 1-Dimensional CNNs.

This is followed by perhaps a second convolutional layer in some cases, such as very long input sequences, and then a pooling layer whose job it is to distil the output of the convolutional layer to the most salient elements. The convolutional and pooling layers are followed by a dense fully connected layer that interprets the features extracted by the convolutional part of the model. A flatten layer is used between the convolutional layers and the dense layer to reduce the feature maps to a single one-dimensional vector.

**The CNN layer may also be followed by a sub-sampling layer to reduce noise in the learned features and this increases the accuracy of the model, but also its complexity.**

## References:


- On Developing a Financial Prediction System: Pitfalls and Possibilities(http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.131.2708&rep=rep1&type=pdf)
- The Efficient Market Hypothesis (https://www.princeton.edu/~ceps/workingpapers/91malkiel.pdf)
- Financial Performance evaluation of Construction Industries (http://www.ijsrp.org/research-paper-0117/ijsrp-p6129.pdf)
- One-step and multi-step ahead stock prediction using backpropagation neural networks(https://www.researchgate.net/publication/271556009_One-step_and_multi-step_ahead_stock_prediction_using_backpropagation_neural_networks)
- Stock Market Prediction using LTSM Recurrent Neutral Network(https://reader.elsevier.com/reader/sd/pii/S1877050920304865?token=18384927099BB2281AB654A2C02B025526B4640ADB6C46D9E63E579505E9E6141FC5C9A49D9433C4EC5A22B5F17B8C24)