A picture containing text

Description automatically generated

**CZ4079**

**FINAL YEAR PROJECT**

**FYP Specific Topics**

Shearman Chua Wei Jie (U1820058D)

FYP Project: SCSE20-0507

FYP Supervisor: Dr Althea Liang

Contents

[Introduction 3](#_Toc59840905)

[Initial Experiments 3](#_Toc59840906)

[Objective 3](#_Toc59840907)

[Experiment Set Up 3](#_Toc59840908)

[Quantifying The Significance For Each Indicator In Terms Impact Of Difficulty In Learning 4](#_Toc59840909)

[Results 5](#_Toc59840910)

[Results (CNN Models) 5](#_Toc59840911)

[Results (LSTM Models) 6](#_Toc59840912)

[Specific Fintech Topics to Tackle 8](#_Toc59840913)

[Past Research on Stock Trading Action Prediction 8](#_Toc59840914)

[Proposed method on Stock Trading Action Prediction 9](#_Toc59840915)

[Financial Crisis Prediction 12](#_Toc59840916)

[Proposed method on Financial Crisis Prediction 12](#_Toc59840917)

# Introduction

Having conducted the initial experimentations on using Deep Learning models for the prediction of stock market direction, this document will go over the conclusions of the initial findings and how various data dimensions affect the difficulty of the deep learning models to learn the generalized pattern of the input data to give a prediction.

We will then look at specific problems that we would like to extend the initial findings to, in the area of fintech. This is so that we can further study the effect of the significance for each particular data dimension in terms impact of difficulty in learning for deep learning model. Also, we will be looking at how we can apply federated learning to these specific fintech problems.

# Initial Experiments

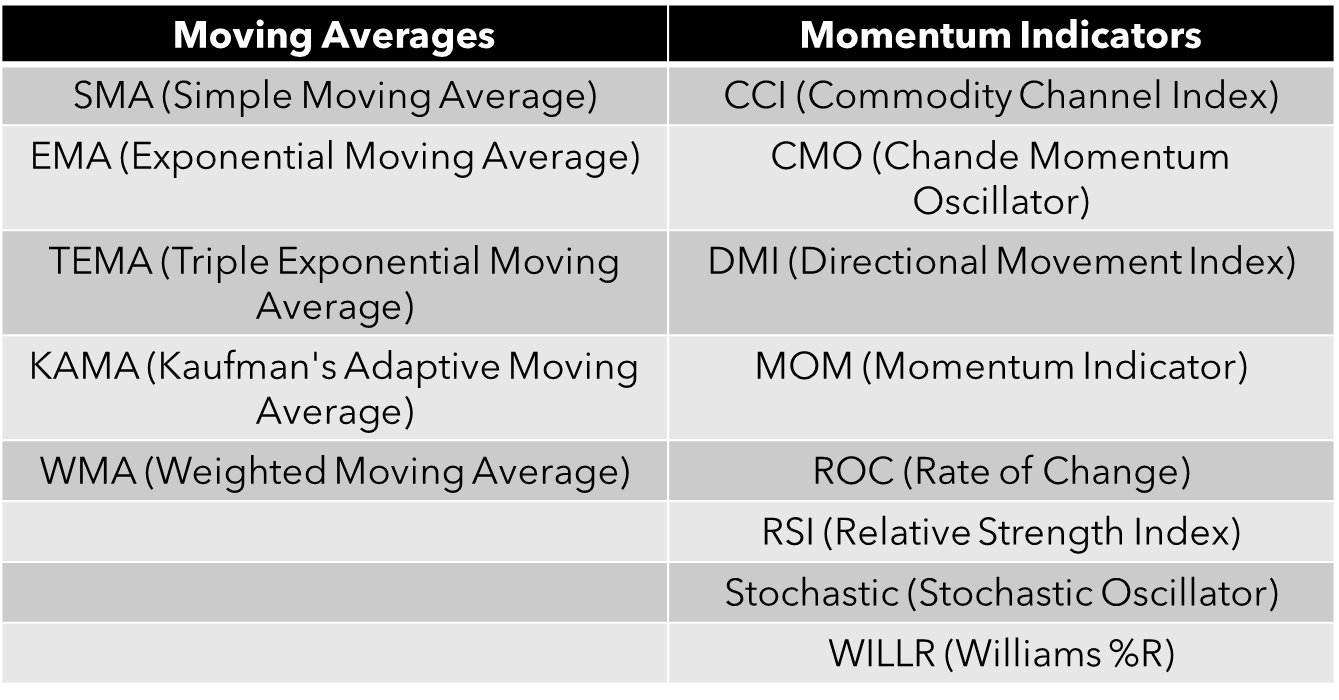
## Objective

For the initial experiments, our deep learning task was to utilize various technical indicators over the past 15 trading days for particular stocks as input to 2 Deep Learning models (LSTM and CNN models) to make predictions on the stocks’ direction, and the output of the models will be a trading signal(buy, hold, or sell) for the stocks.

The primary objective of the initial experiments is to find out for the task of stock market trend prediction, which technical indicator(data dimension) that was more important toward the learning of the stock market trend prediction task amongst all the input technical indicators. In other words, the significance of each particular data dimension in terms impact towards the difficulty in learning of deep learning models.

## Experiment Set Up

From the historical closing, lowest, highest stock prices for each trading day, we are able to derive various technical indicators to be used as input for the deep learning models for the stock market trend prediction task. The technical indicators that are derived can be classified into 2 main groups, technical indicators that are Moving Averages indicators, and technical indicators that are Momentum related indicators. The technical indicators derived are as follows:



Together with the past 15 days Closing Price of the stocks, as well technical indicators, we create an input vector for the LSTM and CNN models.

## Quantifying The Significance For Each Indicator In Terms Impact Of Difficulty In Learning

In order to quantify the significance of the various technical indicators in terms impact of difficulty in learning for the deep learning models, we trained deep learning models where we leave one technical indicator out for each model, for all the technical indicators, and we find out:

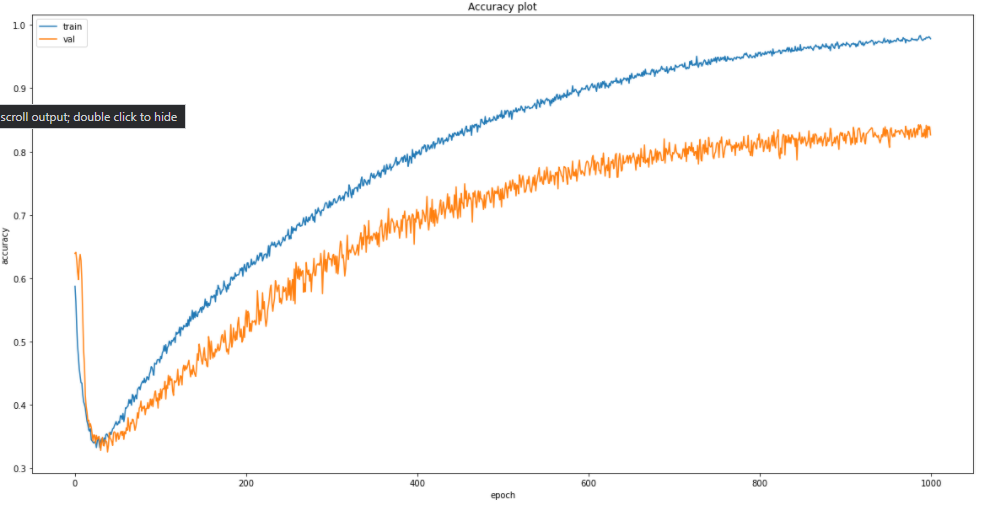
* Final Epoch Validation Loss, Validation Accuracy
* Learning rate for model accuracy (percent per epoch)
* Learning rate for model 10 epochs moving average accuracy (percent per epoch)
* Rate of model accuracy learning jitter
* Rate of model loss learning jitter

For each of the models trained.

We define the Learning rate for model accuracy as the sum of the percentage increase of current training epoch validation accuracy from the previously highest obtained validation accuracy (if the current training epoch validation accuracy is higher than the previously highest obtained validation accuracy) for each training epoch and divide it by the total number of epochs the model is trained (1000 epochs for our experiments). As for most of the models, there is a drop of accuracy to the absolute minima for the first 20 epochs, we take the accuracies for the last 980 training epochs.

**Learning Jitter**

During training, the deep learning models experience learning “jitters”, which are minor fluctuations of accuracies (as seen in the graph below)



Therefore, we want to observe if removing any technical indicators results in more learning ‘jitter’ during training. The rate of learning jitter (%) is defined by:

**learning jitter = (count of minor accuracy drops / number of training epochs) \* 100**

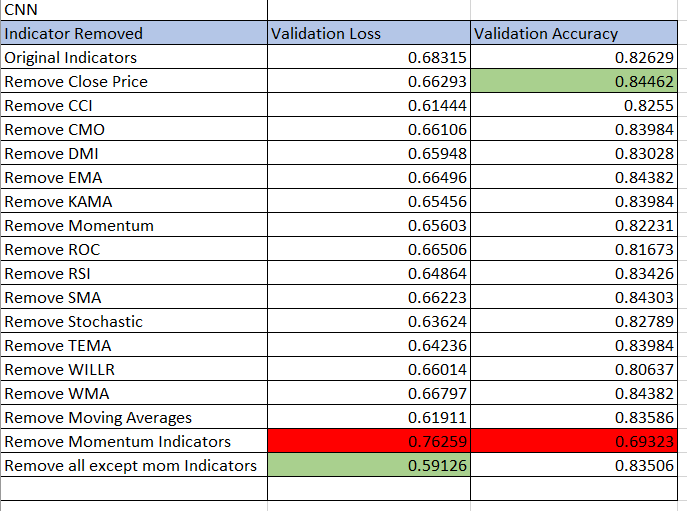
This gives the rate in which learning jitter occurs.

## Results

### Results (CNN Models)

**Validation Loss & Accuracies**

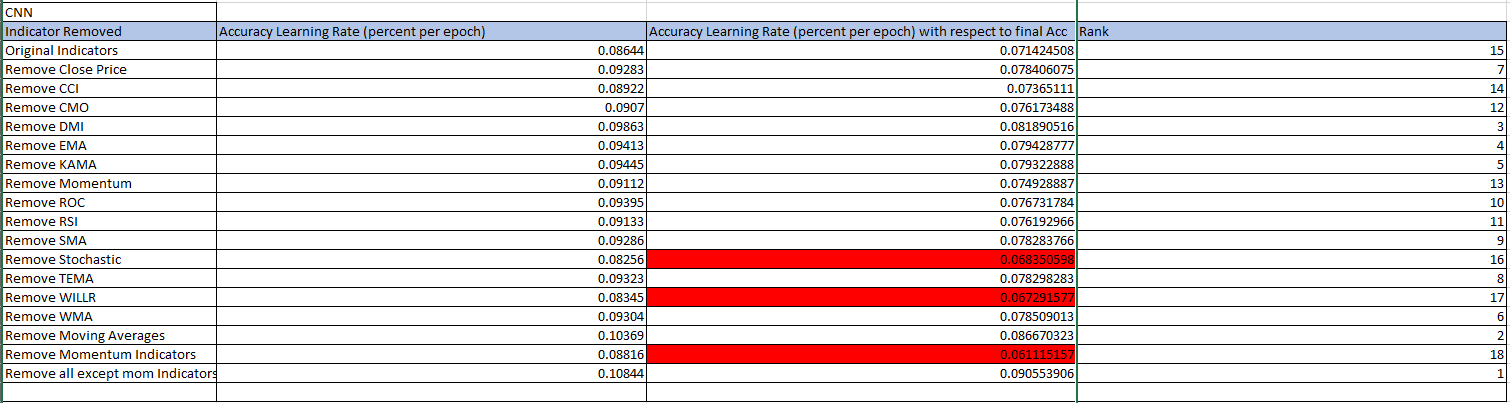
For the CNN models that were trained, the validation accuracies of the models were around the same when we remove only a single technical indicator from the input data. Therefore, we also experimented with removing all the technical indicators from one of the 2 main groups of indicators, technical indicators that are Moving Averages indicators, and technical indicators that are Momentum related indicators to train 2 additional models. From the experiments, we found that when we remove all the Momentum related indicators, the validation accuracy of the model drops quite significantly, from 80 plus percent to 60 plus percent. Also, the validation loss is also the highest amongst all the models trained when we remove all the Momentum related indicators.



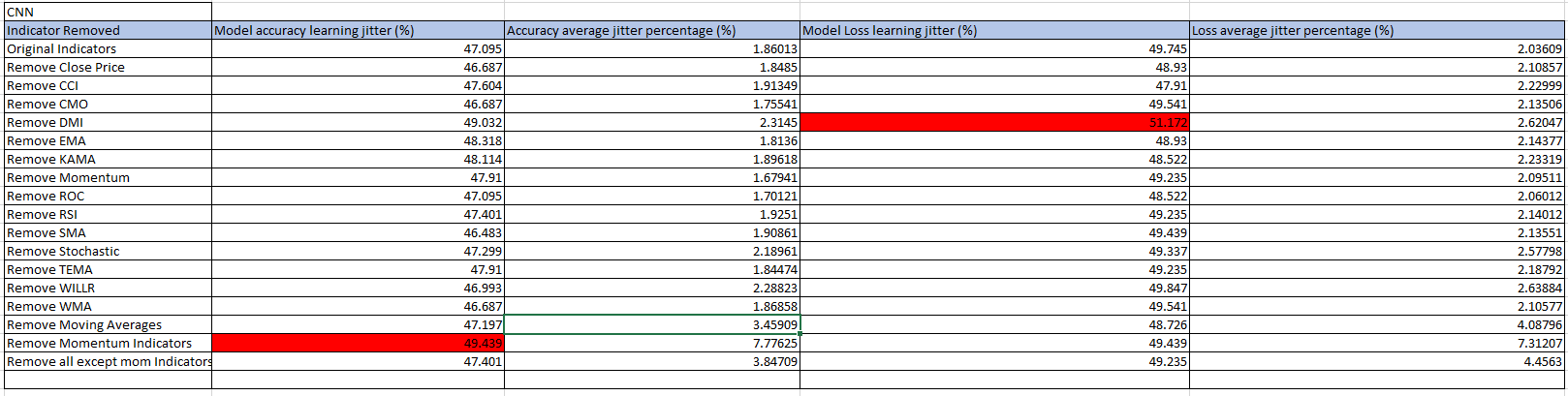
**Learning Rate for Model Accuracy**

We also found out from the experiments that when we rank the models according to the Learning rate for model accuracy, the bottom 3 ranked models were:

* The model with Stochastic Indicator removed
* The model with WILLR Indicator removed
* The model with all Momentum Indicators removed



**Learning Jitter**



For the effects of technical indicators on Learning Jitter during training, the results are not very conclusive when comparing the Learning Jitter of models that have only 1 indicator removed. However, removing momentum related indicators resulted in the most learning jitter and the highest average jitter percentage deviation.

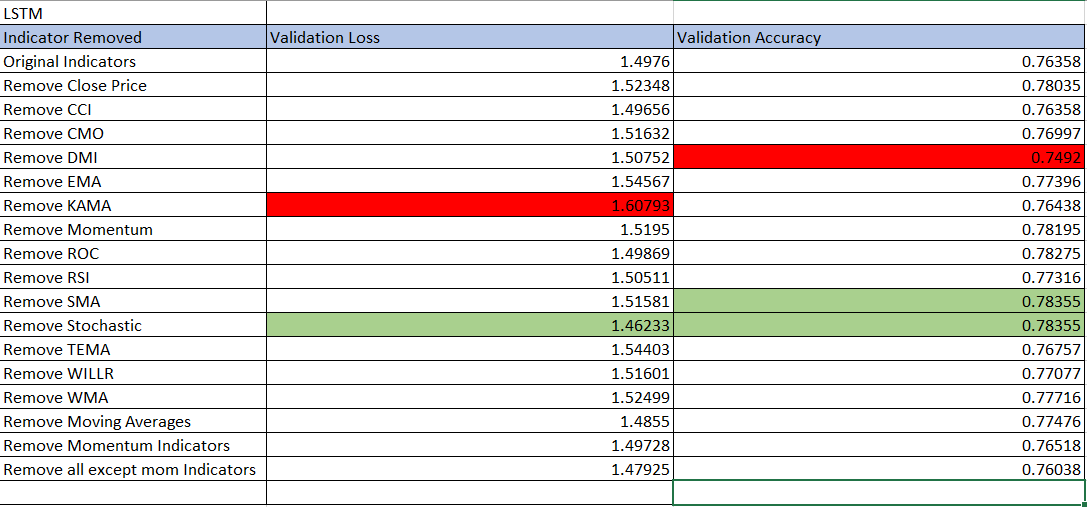
**Conclusion**

Therefore, we can conclude that for the impact of technical indicators on the difficulty of learning in CNN models, Momentum indicators have more significant impact on the difficulty of learning in CNN models and Moving Averages and Closing Prices does not help much in learning of the prediction task for the CNN models. The model that performed best, is the model using only momentum related indicators. As for Learning Jitter, in general, removing more indicators causes more learning jitter.

### Results (LSTM Models)

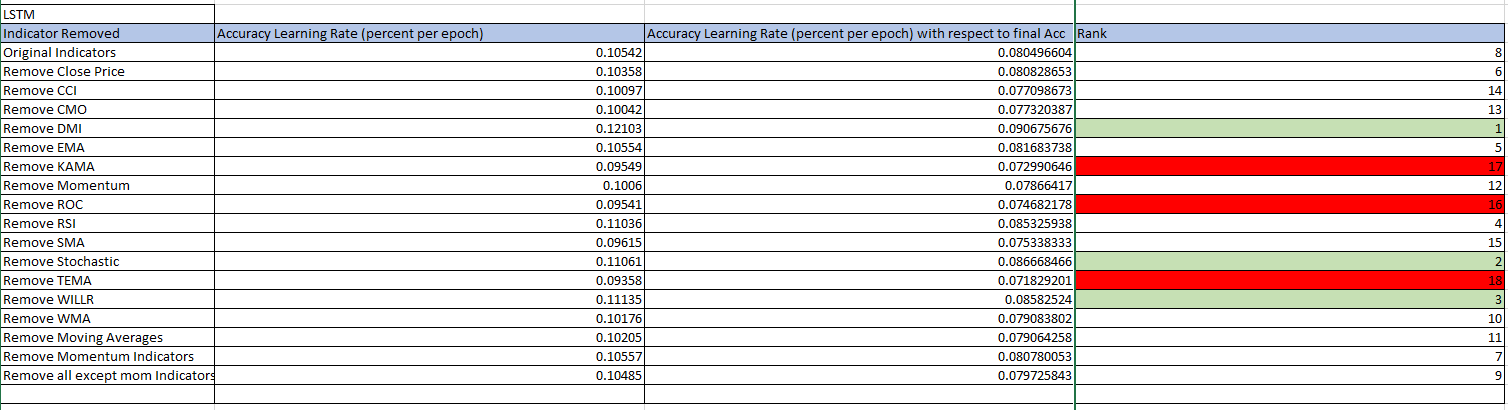
**Validation Loss & Accuracies**

For the LSTM models that were trained, when we removed single technical indicators, the results were not very conclusive, where the validation accuracies and validation losses of the models do not really differ that much from one another. Even when experimented with removing all the technical indicators from one of the 2 main groups of indicators, technical indicators that are Moving Averages indicators, and technical indicators that are Momentum related indicators to train 2 additional models, the results were fairly similar to those of the models with only one indicator removed.

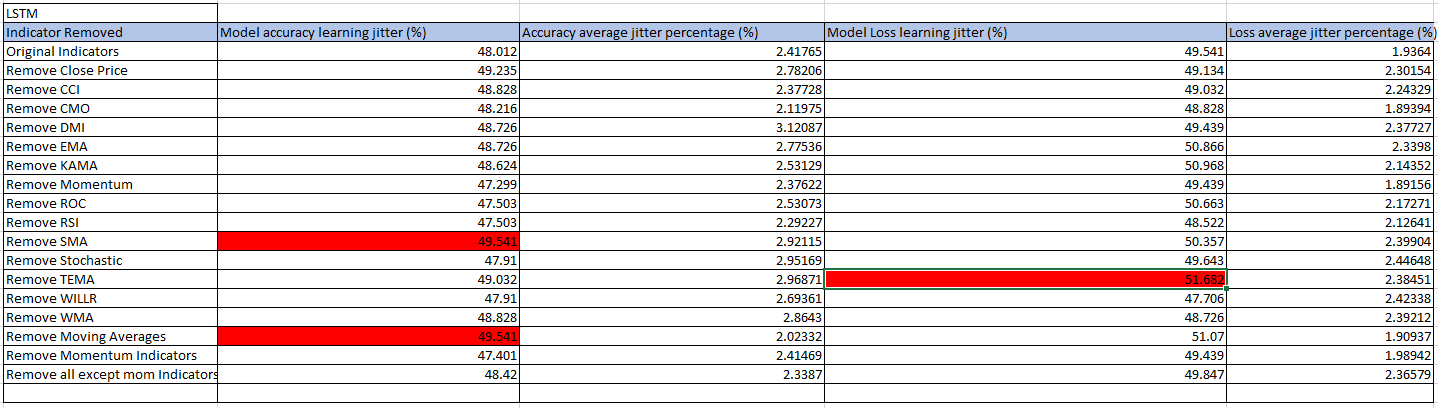


**Learning Rate for Model Accuracy**

We also found out from the experiments that for LSTM models, the results are not as conclusive as the CNN models results where we are not able to determine if momentum related indicators or moving average related indicators impact the learning of the models more.



**Learning Jitter**



For the effects of technical indicators on Learning Jitter during training, the results are not very conclusive. However, removing moving average related indicators resulted in the more learning jitter generally as compared to removing momentum related indicators.

**Conclusion**

Therefore, we can conclude that for the impact of technical indicators on the difficulty of learning in LSTM models, compared to the CNN models, we were unable to determine which indicators are more significant to the impact on the difficulty of learning for LSTM models. In addition, the model with the best learning rate is the model with all the indicators included.

# Specific Fintech Topics to Tackle

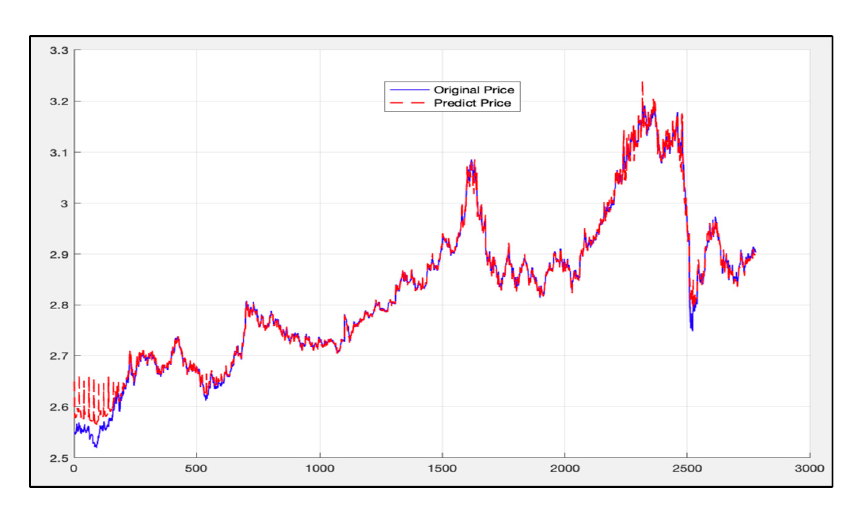
With the conclusions and the results that we have obtained for the initial experiments, we would then want to extend our project to solve specific Fintech problems as well as to further conduct our experiments on the impact of data variation on the difficulty of learning for Deep Learning models.

## Past Research on Stock Trading Action Prediction

There has been a widespread use of Deep Learning models such as LSTM (Long Short-Term Memory) RNN models, and CNN (Convolutional Neural Network) models for the purposes of stock market related applications in many past research. LSTM models have been used in many past research for the purpose of predicting the **next trading day stock price** of stocks and CNN models have been used for the purpose of predicting the trading action (to buy, hold or sell the stock) to perform on the stocks based on the input data given.

However, the way the models are developed, are not practical for real world applications as well as the fact that we have no definitive answers as to which data dimensions/factors are the most impactful towards helping Deep Learning models learn a generalized prediction pattern for stock trading applications.

For example, LSTM models have been used to predict the stock prices of the next trading day based on the past **n** number of days closing prices of the stock. While in many research paper, the results obtained by these models seems promising, the prices predicted are not really useful to stock traders to decide whether to perform a trading action on the stock. It just informs the stock trader the estimated stock price for the next trading day. In addition, the predicted stock prices are usually “lagging” the actual stock prices which can be seen if the predicted and actual prices are plot on the same graph. What this means that when there is a change in the stock direction ( rising-to-falling or falling-to-rising), the predicted prices can only reflect the change in stock direction a few trading days later. Therefore, stock traders are not able to make use of the predicted stock prices to predict changes in stock directions which can be helpful for traders to make traders. This renders LSTM models to predict the stock prices of the next trading day not very useful for stock trading.



**Figure from Yujie Fang, Juan Chen, and Zhengxuan Xue, Research on Quantitative Investment Strategies Based on Deep Learning showing “lagging” predicted prices**

For previous research using CNN models, researchers uses CNN to determine the "Buy", “Hold”, and "Sell" points in stock prices using 15 different technical indicators with different time intervals and parameter selections for each daily stock price time series to create images.

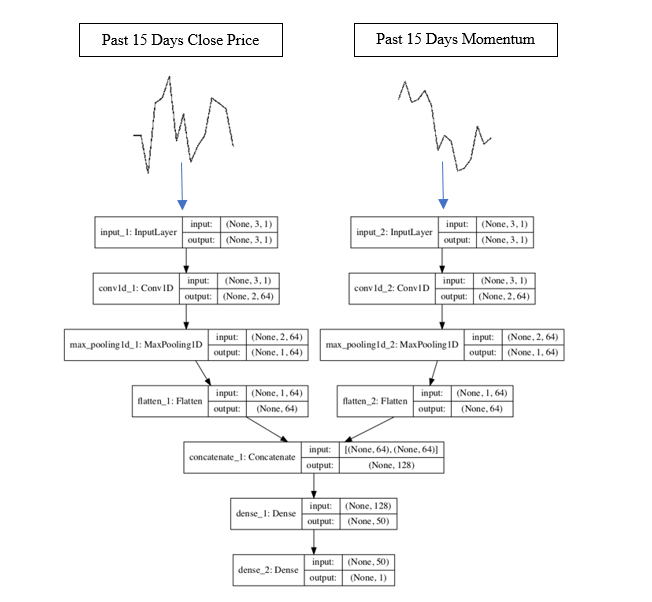
The technical indicators for each day, RSI, Williams %R, WMA, EMA, SMA, HMA, Triple EMA, CCI, CMO, MACD, PPO, ROC, CMFI, DMI, and PSI values for different intervals (6 to 20 days) are calculated and for each day a 15x15 image is generated by using 15 technical indicators and 15 different intervals of technical indicators.

From one of these research, we can observe that the precision and recall of the model is not very high for the “buy” and “sell” trade action labels. This means that the model will not be very practical for use in a real-life trading scenario as how stock trades profits is by making “buy” and “sell” . However, with that being said, the model can still be profitable if most of the predicted “buy” points are cost lesser than most of the predicted “sell” points.

A plausible reason for the CNN model not performing well when predicting the “buy” and “sell” trade action labels is the fact that the input data contains too many data dimensions and we do not know which data dimensions are actually useful in helping the CNN model learn the prediction of stock trade action labels. In addition, the way the data “image” is created, compresses each technical indicator past 15 days data points as a 1-D vector, and we know that Convolutional networks were inspired by biological processes in that the connectivity pattern between neurons resembles the organization of the animal visual cortex where it may be hard for the CNN model to “visualize” the 1-D vectors of the 15 technical indicators, making it hard to learn the “patterns” in the technical indicators to learn them efficiently to make predictions on stock trade action labels.

## Proposed method on Stock Trading Action Prediction

For the specific problem of Stock Trading Action Prediction, we propose using a multi-headed 2D CNN model used to determine the "Buy", “Hold”, and "Sell" points in stock prices. Instead of creating day a 15x15 image generated by using 15 technical indicators and 15 different intervals of technical indicators, like in past research and for our initial experiments, each “head” of the CNN model will take in a 128 x 128 image of a graph plot of the past 15 days data points of a single technical indicator (e.g. past 15 days closing price, past 15 days momentum), and the image will pass through convolutional layers and pooling layers before all the “heads” of the models are concatenated into a single “tail” with dense layers and finally give the "Buy", “Hold”, and "Sell" points labels as outputs.



**Multi-headed CNN Model**

In addition, in order to find out the significance of each particular data dimension in terms impact towards the difficulty in learning of deep learning model, we can fix the Multi-headed CNN model as a two-headed CNN model where we keep the input to one of the “head” of the model constant and vary the other “head’ of the CNN model with various technical indicators debt, credit rating, sentiments and etc.

Also, the technical indicators to be used for Stock Trading Action Prediction can be further classified as Moving Averages or Momentum related indicators, Volume indicators, and lagging or leading indicators.

|  |  |  |
| --- | --- | --- |
| Indicator | Moving Average/ Momentum or Volume | Leading or Lagging |
| SMA (Simple Moving Average) | Moving Average | Lagging |
| EMA (Exponential Moving Average) | Moving Average | Lagging |
| TEMA (Triple Exponential Moving Average) | Moving Average | Lagging |
| WILLR (Williams %R) | Momentum | Leading |
| RSI (Relative Strength Index) | Momentum | Leading |
| Stochastic (Stochastic Oscillator) | Momentum | Leading |
| On-balance volume (OBV) | Volume | Leading |
| Parabolic SAR | Moving Average | Leading |
| Chaikin A/D Oscillator | Volume | Leading |

In addition, in order to ensure that the “buy” and “sell” points are profitable in real life stock trades, we will need to re-label the output labels of the data from our initial experiments to take into account trading fees. This is done by labelling for each 15-day trading window, if the highest closing price in that window is higher than 0.275%(trading platform cost is 0.275% of trade amount) increase of the lowest closing price in that window, we label the lowest closing price as “buy” and the highest closing price as “sell”. This is to ensure that when we make a trade, we do not make a loss due to the fact that the trading cost is more than that of the trade profit.

**Integration with Federated Learning**

In the stock market, stocks can vary from one another in terms of trade volume each trading day, the volatility of the stock prices, as well as the market Cap of the underlying company. All these factors make trading strategies from one stock to another very different. For example, for a Blue Chip stock that has a large market Cap, the stock tends to be more stable and follows the general trend of the global stock exchanges as well as the stock exchange the stock is traded in. On the other hand, a stock that has a small market Cap will tend to be more volatile and traded more frequently as swing traders frequent sell and buy these stock to make a profit out of them, causing their trade volume to be very high as well.

Therefore, a Deep Learning model trained on one particular stock may not fit the need of all the stocks in the stock market. Using federated learning, we can first train a general model and the weights of the models are then transferred to all the other devices participating in the federated learning cluster. With federated learning, each local device can then train the model based on the specific stocks trade on that device and allow the weights of the local model to be changed according to the learning pattern of the stocks traded on the local device. After which, the weights of the local device are then sent to the main general model, and if the weights improves the accuracy of the general model, the weights of the general model will be updated as well.

## Financial Crisis Prediction

Financial Crisis is a very serious problem that can plague economies around the world. A global financial crisis can result from a series of local or/and regional market shocks, which can lead to a worldwide economic crisis due to the interconnectedness of the financial markets. The recent Covid-19 pandemic caused the world to go into a global recession in the year 2020 from late February 2020 and showed signs of recovery toward the end of September 2020.

Therefore, if we are able to forecast Financial Crisis **n** days in advance before it happens, we can prepare for it and cushion the impact of the Financial Crisis so that the damage to the economy is as minimal as possible.

## Proposed method on Financial Crisis Prediction

In order to perform Financial Crisis Prediction, we must collect information on various financial indicators from a number of different sources and databases. We will be focusing on liquid markets, where the transmission of extreme events is better depicted in the pricing patterns. Specifically, the data will include the Major Stock Indices around the world, the yield of the 10-year government bond of various countries across the globe, the exchange rates of 18 currencies against the United States dollar, as well as additional financial indices such as Oil, Gold. These selected financial indices should include the most sensitive market variables, covering the three most important financial markets, namely America, Asia, and Europe.

In addition, usually the events happening around the globe as well as the sentiments towards them will affect the stability of the financial markets, for example the outbreak of the Covid-19 pandemic. The GDELT Project, or Global Database of Events, Language, and Tone, created by Kalev Leetaru of Yahoo! and Georgetown University, along with Philip Schrodt and others, describes itself as an initiative to construct a catalogue of human societal-scale behaviour and beliefs across all countries of the world, connecting every person, organization, location, count, theme, news source, and event across the planet into a single massive network that captures what's happening around the world, what its context is and who's involved, and how the world is feeling about it, every single day. In the GDELT data, there is the **GoldsteinScale** column s assigned a numeric score from -10 to +10,capturing the theoretical potential impact that type of event will have on the stability of a country. Therefore, we can also incorporate the use of the GDELT data set as an input factor for Financial Crisis Prediction.

**Data Dimensions Identified:**

* Major Stock Indices Data
* Yield of 10-year government bonds of various countries
* Exchange rates of 18 currencies against the United States dollar
* Additional financial indices such as Oil, Gold
* GDELT data

With the data that will be retrieved, we will then create a 15-day time series data tensor which will contain the various data dimensions for the past 15 days. This will be then used to create 2 models predict if a Financial Crisis will occur in the next 1 week and predict if a Financial Crisis will occur in the next 2 weeks, respectively. The output of the model will be a binary class where 1 represents an occurrence of Financial Crisis and 0 represents a non-event.

We will be using the LSTM Deep Learning model for this prediction task as LSTM models are better at capturing time-series information and that should be able to aid the task of Financial Crisis Prediction using past 15 days data inputs.

**Integration with Federated Learning**

The model can be extended to predict the occurrence of Financial Crisis within a particular country/region. Therefore, we can make use of Federated Learning to create Local models, each of which is able to predict the occurrence of Financial Crisis within a specific country/region. This is due to the fact that the occurrence of Global Financial Crisis may not be as common as compared to contained country or regional Financial Crisis.