Small Cap Biotech Stock Price Movement Analysis

Tejas Krishna Reddy
EECE, College of Engineering (COE)
Northeastern University
Boston, United States
krishnareddy.t@northeastern.edu

Phanindrakumar Chintapalli
EECE, College of Engineering (COE)
Northeastern University
Boston, United States
chintapalli.p@northeastern.edu

***Abstract*— Investing in Biotech stocks is a very risky task due to its high volatility and risk. However, it's the most profitable investment if it's done right. Small capital biotech companies solely depend on their success in clinical trials towards final FDA approval. Since, this final FDA results are a major news for the company, the stock price shoots up/down based on the news released. Many researchers have proved the fact that there is a significant pattern that can be recognized during this time period. Therefore, in this project we wanted to learn that pattern through a machine learning algorithm and check if we are able to predict the best time periods to buy and sell the stock for maximum profits. We created a dataset with 177 events where the drug was approved and extracted their stock price movement during the period of interest. Also, many lead technical indicators were attached to these as additional features that could help the model classify better. From this, an LSTM model was trained that predicted the buy and sell signals with ~87% accuracy.**

Keywords—Biotech, Stock price, LSTM, Classification, Trading Strategy, Fintech.

I. Introduction

Predicting stock price movements is an intricated task. No algorithm has ever been developed to accurately predict the movement of stock price of a company. The main reason for this is that the share price depends on a variety of factors such as discounted cash flow's, companies popularity, its demand in the market, logistics, supply chain process, profits/losses, news released and the public's perception/ interpretation of the released news. Along with these, how big investors such as hedge fund companies involve changes the course of price drastically. Developing machine learning models that work for every company in NASDAQ is not possible since every company is fundamentally different from each other, for example stock movement for a new product announcement for Apple is very different from Lutron Electronics. Hence, majority of quant developers end up developing separate models for each company.

Using ML models in stock market predictions is still not a feasible idea for the above two reasons explained where one would have to factor in multiple dozens of factors develop a feature set which again changes dramatically from one company to another. Hence, we wanted to select a group of stocks that would have very similar feature set to each other and whose feature set is as simple as possible compared to other corresponding sectors such that we could develop one model that could predict their movements to a fairly good accuracy.

Small-capital biotech companies are small drug discovery companies whose fate lies in the success of the clinical trials through phase 1/2/3 and eventually final approval by FDA into the market. These companies stock price depends on the news related to its clinical trials and other investors reaction to the news.

Many researchers have proven that there is a significant pattern in stock price movement before and after the event of the final FDA approval date. Our aim in this project was capture this pattern using machine learning models to later use that movement to identify best possible situations to buy and sell the stocks for realizing maximum profits from these stocks. (Assuming we had an institution that the clinical trial would pass FDA in the near future).

We created a dataset with 177 events where the drug was approved by the FDA and extracted their stock price movement during six months prior to the event date and six months after the event date. Also, many lead technical indicators were attached to these as additional features that could help the model classify better. Technical indicators such as Moving Average Convergence Divergence (MACD), Relative Strength Index (RSI) are very popular and powerful trend recognizers. We also calculated the Velocity (first derivate of close price), Acceleration (2nd derivative of close price) and Jerk (3rd derivative of close price). From this, an LSTM model was trained that predicted the buy and sell signals with ~85% accuracy.

II. Literature Review:

There is no current research in the exact same project, whereas individual parts of the project, namely analytical pattern recognition in small cap biotech stocks movement during the FDA approval period and general stock price movement predictions using Machine Learning have been very actively researched.

The paper [1] proves that positive results from phase 1/2/3 have a significant raise in stock price. The raise is more for FDA Approval > Phase 3 > Phase 2 > Phase 1 respectively. Whereas, for a negative outcome there is a dip of approx. 1.7% of the price irrespective of which phase the results belong to. Paper [2] compounds all the cumulative abnormal returns (CAR's) for all stocks similar to our analysis from -120 days to +120 days before and after the event. Below graph from the paper summarizes their findings very clearly.

Paper [3] shows us how many hedge fund companies trust this analysis and place their bets on clinical trials in later stages, since they have a definitive raise if the results are positive.

Paper [4] explains how an encoder-decoder Long-Short-Term Memory network is useful in determining the stock price movements when modelled for individual companies. Effectiveness in understanding the different type of features that can be gathered while creating a dataset which can help model determine the movement better are clearly explained in [5].

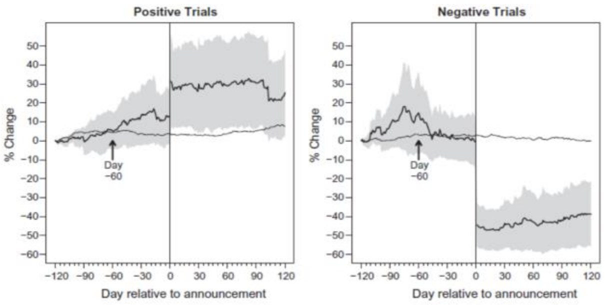

Fig. 1. Stock Price reaction to positive and negative results announcement for multiple stocks compounded and normalized. (Source: [3]).

III. Dataset and Features:

We developed a web crawler to scrape all the historic events when the drug was approved by FDA from [6] website. Therefore, we obtain around 2137 events with their date, result and corresponding companies stock ticker. Currently, our aim is to develop a model only for the approved results. Hence, we filter out the clinical phase results and rejected trials. We remain with 597 events. Now, we filter out all the large capital companies, whose market capital is greater than 2 billion dollars. Therefore, we finally have 177 small cap biotech approved events in our database.

Using the stock tickers in the 177 events, we download their daily OHLCV (Open, High, Low, Close, Volume) data for the required companies from January 1st 2000 to August 1st 2020 from yahoo finance website [7] using pandas data-reader object.

We use closing price of the data as our main feature. Inorder to smoothen the curve and extract the trend line and remove the noise we apply Triangular Exponential Moving Average (TEMA) algorithm. Triangular EMA is the EMA of the EMA of closing price. We currently use past 6 days data to smoothen the price graph. The number 6 was chosen after multiple trial and error experiments in a way it smoothens the graph but does not lag the information too much to mislead the ML algorithm we build later on. EMA is defined as below:

$$\mathrm{EMA}_{\mathrm{Today}} = (\mathrm{Value}_{\mathrm{Today}} * ((1+\mathrm{nDays}) / \mathrm{Smoothing})) + \mathrm{EMA}_{\mathrm{Yesterday}} * (1-((1+\mathrm{nDays}) / \mathrm{Smoothing}))$$

We also add two of the most popular and proven trend technical indicators, namely Relative Strength Index (RSI) and Moving Average Convergence Divergence (MACD) which is divided into 3 features Signal, MACD and difference between them both.

RSI for closing price is defined as:

$$RSI = 100 - [\frac{100}{1 + \frac{Avg\ previous\ gain}{Avg\ previous\ loss}}]$$

MACD and Signal are defined as:

$$MACD = 12\ Day\ EMA - 26\ Day\ EMA$$

$$Signal = 9\ day\ EMA\ of\ MACD$$

We also added velocity, acceleration and jerk from the closing price time series information as features which are defined as below:

$$Velocity = \frac{d}{dx}(Closing\ Price)$$

$$Acceleration = \frac{d}{dx}(\frac{d}{dx}(Closing\ Price))$$

$$Jerk = \frac{d}{dx}(\frac{d}{dx}(\frac{d}{dx}(Closing\ Price)))$$

Therefore, we extract 300 days before the event and 300 days after the event stock OHLC information for all 177 events in the database and create TEMA, Volume, MACD, Signal, RSI, Velocity, Acceleration and Jerk as additional features for the modelling purpose.

Now, we also use python's Peak-Detect [8] package to manually identify the peaks with a lookahead of 15 days. Basically, we identify the best points to buy and sell in the smoothened TEMA curve if 15 days of future was already defined. Now, the recognize the buy and sell points recognized by the algorithm and use them as our ground truth labels for the developed dataset.

Therefore, we finally have features and labels generated. We have a combined 78,130 samples, with 10 features and a target label. We have 1303 buy signals and 1289 sell signals in the recorded data.

IV. Approach

Firstly, we normalize the price for all stocks to 1. Since, every stock has its own base price, in order to compare and extract patterns it is important to convert their price to a percentage of its original price at day 1. The features Volume and TEMA are highly tail skewed and hence we apply exponential log normalization to both those features. Log transformation is a data transformation method in which it replaces each variable x with a log(x). The choice of the logarithm base is usually left up to the analyst and it would depend on the purposes of statistical modeling. In this project, we will focus on the exponential log transformation. When our original continuous data do not follow the bell curve, we can log transform this data to make it as "normal" as possible so that the statistical analysis results from this data become more valid.

Later, we do standard scaling for all the features in the dataset. Standardization is another scaling technique where the values are centered around the mean with a unit standard deviation for feature set X:

$$X = \frac{X - \mu}{\sigma}$$

Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample. The procedure has a single parameter called k that refers to the number of groups that a given data sample is to be split into. As such, the procedure is often called k-fold cross-validation.

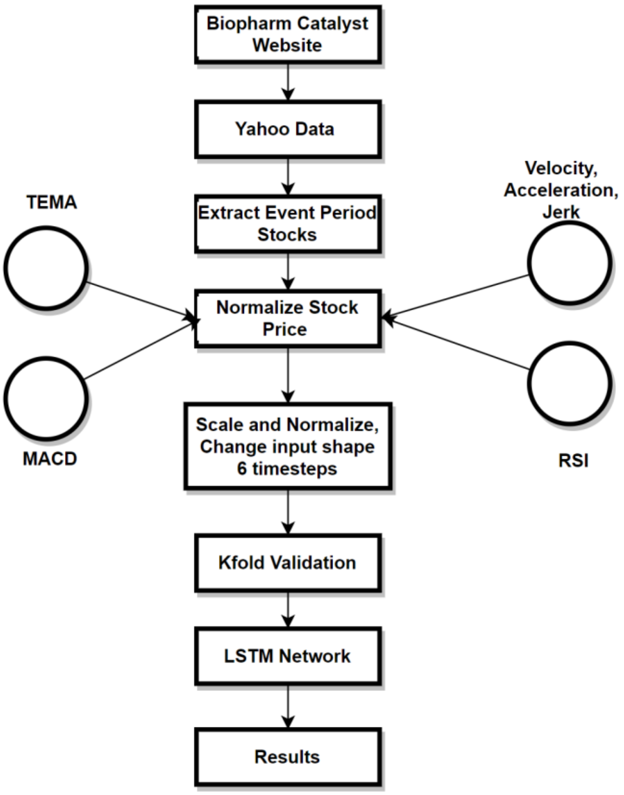
Biopharm Catalyst Website
Yahoo Data
TEMA
Velocity, Acceleration, Jerk
Extract Event Period Stocks
Normalize Stock Price
MACD
RSI
Scale and Normalize, Change input shape 6 timesteps
Kfold Validation
LSTM Network
Results

Fig. 2. Block Diagram respresenting the approach taken.

Cross-validation is primarily used in applied machine learning to estimate the skill of a machine learning model on unseen data. It divides the dataset into k-folds where the model is trained on K-1 folds and tested on the unseen fold. That is, to use a limited sample in order to estimate how the model is expected to perform in general when used to make predictions on data not used during the training of the model. It is a popular method because it is simple to understand and it generally results in a less biased or less optimistic estimate of the model skill than other methods, such as a simple train/test split. The sanity of the model can be verified with these results.

We then send this data into a Recurrent Neural Network for forecasting. Recurrent Neural Networks suffer from short-term memory. If a sequence is long enough, they’ll have a hard time carrying information from earlier time steps to later ones. During back propagation, recurrent neural networks suffer from the vanishing gradient problem. Gradients are values used to update a neural networks weight. The vanishing gradient problem is when the gradient shrinks as it back propagates through time. At the same time if the gradient becomes too big due to multiple matrix multiplications, then it explodes the model due to memory constraints. Hence, through this, the initial weights stop being updated if the model is big. Inorder to avoid this issue, we send this data to an LSTM network. They have internal mechanisms called gates that can regulate the flow of information. These gates can learn which data in a sequence is important to keep or throw away. By doing that, it can pass relevant information down the long chain of sequences to make predictions. Almost all state-of-the-art results based on recurrent neural networks are achieved with this network. We use CuDNNLSTM’s [9] in the script, which is CUDA enabled python library that is much faster than a regular LSTM in training.

V. Experiments and Results

Firstly, we check for a recognizable pattern in the data collected to prove the sanity of the hypothesis. Therefore, we take the average for each day (-50 days to +50 days of the event, where 0th day was the day of the event) after normalizing the prices for all the stocks in our data to 1. In the below Fig 3.(a). we look at all the 177 stock price movements from day -50 to 50 and in Fig 3(b) we can clearly observe a 8% rise in stock price during the event of approval result and then an immediate market cool down and fall of the prices. This happens due to the market realization of the stock where huge investors withdraw their cash as the stock price is shooting to the highest possible price.

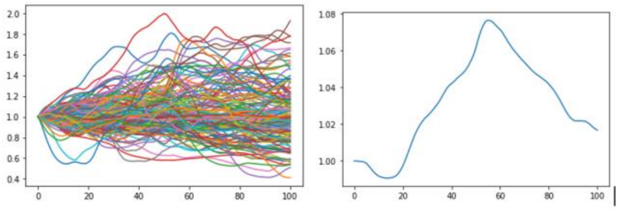

Fig. 3. (a). All stocks price movement (b) Average Stock movement

An LSTM layer was developed as shown in Fig 4. Initially the model faced overfitting problem, which was then overcome by sampling the data equally and also by adding drop out layers in the network to minimize the number of weights in the overall network. We tried to keep the model as a simple model since the number of (Buy, Sell) samples in the data were considerably low and did not want to fit a huge layer and face underfit since all the weights could not be balanced with minimal back-propagation. We used 6 time-steps at a time, trying to provide the model the past 6 days features to predict buy/sell signal for the current day. Providing much older data would confuse the model more since acceleration, velocity and RSI are slightly leading

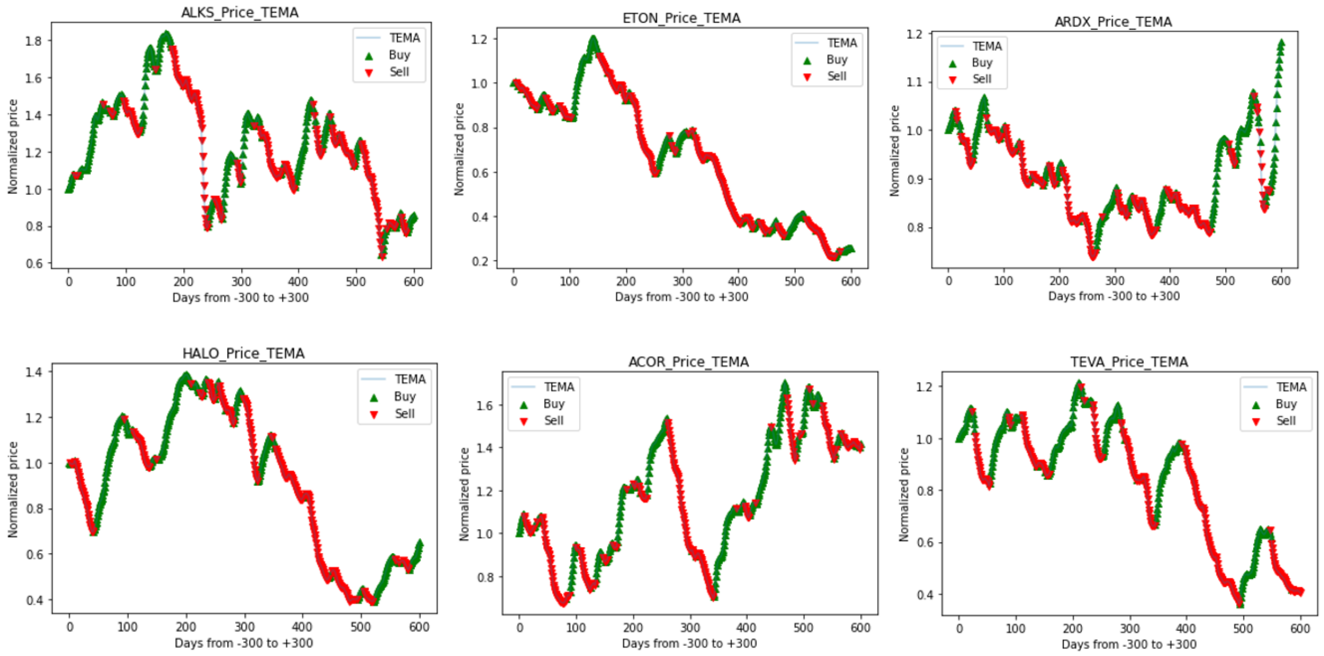

Fig 5: Visual Representations of Buy and Sell Predictions for Unseen Market Data by the LSTM Model.

indicators which cannot function well when a long tail is considered. We currently use a batch size of 30 since, each stock will have 20 back propagations to learn the algorithm (600 days per stock).

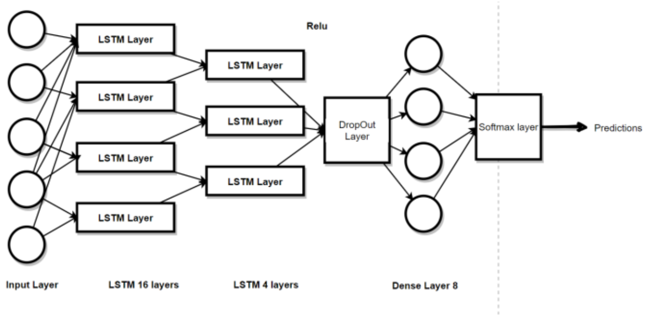

Fig 4. LSTM layers

To analyze the performance of the stock, we look at the average training and testing accuracies from 10-fold validation and from those we can determine that there is no overfitting or underfitting issues in the model. We can also observe a good F1 score along with precision and recall scores. Precision score here, represents the model's ability in identifying buy signal to a good accuracy and recall shows how good are we able to predict a sell signal. The performance matrix can be seen in table 1.

Training Accuracy	Testing Accuracy	AUC Score	F1 Score	Precision	Recall
0.8913	0.8726	0.9754	0.9319	0.9616	0.9040

Table 1: Performance Matrix

Comparatively model is doing better in identifying buy signals than sell signals, which can also be visually seen in the examples in Fig 5.

Confusion Matrix	Buy	Sell
Buy	1242	47
Sell	125	1178

Table 2: Confusion Matrix

VI. Contributions

Tejas Krishna Reddy has been passionate about stock market analysis and building trading strategies for the last couple of years. Hence, his idea on LSTM modelling to predict small cap bio-tech stock movements was taken for the project. His knowledge on, data sources, technical indicators and Machine Learning helped to precisely put together the feature set into place. Before finalizing the current feature-set used for the analysis, several trial and error methods were experimented to understand the effect of features and different indicators in making the models classify better. Preprocessing the data by normalizing, identifying the pattern to prove the sanity of the hypothesis were done by Tejas.

Once, the data was put together, Phanindra Kumar Chintapalli played a vital role in Log-Normalizing the data, Scaling and developing LSTM layers. While developing and testing the final model, initially there was a huge overfit problem that generalized the model too much. Hence, continuous efforts were made by both Tejas and Phanindra in successfully optimizing the model. Meanwhile, in the obsession of making this project a success both of us dwelled deep into other algorithms, techniques and features that could make this better and did implement majority of them to test their validity. We also, divided our work equally in preparing the report and the analysis PowerPoint for the presentation.

References

[1] Chen, Y. J., Feng, Z. Y., Li, Y. P., & Huang, H. W. (2020). The economic consequences of US FDA new drug approvals: evidence from Taiwan pharmaceutical and biotech companies. *Innovation*, 1-21.

[2] Sumadi, F. (2016). Event-Based Biotechnology Stock Price Movement: Valuing Success and Failure in Biotechnology Product Development.

[3] Hwang, T. J. (2013). Stock market returns and clinical trial results of investigational compounds: an event study analysis of large biopharmaceutical companies. *PloS one*, *8*(8), e71966.

[4] Ding, G., & Qin, L. (2020). Study on the prediction of stock price based on the associated network model of LSTM. *International Journal of Machine Learning and Cybernetics*, *11*(6), 1307-1317.

[5] Mwamba, J. W. M. (2020). Modelling stock market behaviour with machine learning techniques.

[6] https://www.biopharmcatalyst.com/calendars/historical-catalyst-calendar

[7] https://pandas-datareader.readthedocs.io/en/latest/remote_data.html

[8] https://pypi.org/project/peakdetect/

[9] https://www.tensorflow.org/api_docs/python/tf/compat/v1/keras/layers/CuDNNLSTM