Brandon Peh: bcp39@stanford.edu
Ren Hao: renhao@stanford.edu (not enrolled in the class)

Deep Learning to Detect Pump-and-Dump

Introduction

We have decided to narrow the scope of our project to focus on the accurate and timely prediction of sudden stock price increases attributed to "pump-and-dump" schemes. This paper defines a *sudden increase* as a price jump of more than 50% across a five-day trading period. A pump-and-dump scheme is the illegal act of an investor or group of investors promoting a stock they hold and selling once the stock price has risen following the surge in interest as a result of the endorsement. In general, these scheme go through the below phases:

- ***Phase 1:*** Stocks which have (i) small market capitalization, (ii) low trading volume and (iii) volatile fundamentals are selected by scammers to be "pumped and dumped" as their prices can be more easily manipulated

- ***Phase 2:*** Scammers incrementally acquire long positions in these targeted stocks in a manner so as to avoid detection, i.e. no significant price movements

- ***Phase 3:*** Scammers trigger sudden price increases through small but expensive buy orders, alongside an aggressive stock promotion campaign; "pump"

- ***Phase 4:*** Scammers take profit by selling their long positions once stock price has gone up substantially; "dump"

We wish to use deep learning to identify stocks which are undergoing Phases 1 - 3 in the universe of US NYSE and NASDAQ listed securities. This paper hypothesizes that it is possible to adequately describe Phases 1 - 3 using non-linear interactions of four time-varying covariates: prices, trading volumes, number of outstanding shares and earnings.

In Phase 1, many metrics which scammers could use to screen for potential pump-and-dump stock targets are derivatives of these four covariates. Example:

Pump-and-dump screening metrics	***Base fundamental metric***
Market capitalization	Number of outstanding shares, price
Price-to-earning ratio	Price, earnings
Earnings per share	Number of outstanding shares, earnings
Trading volume	Trading volume

In Phase 2 - 3, the variation of price and volume across time could potentially reveal the actions of scammers. Example:

Brandon Peh: bcp39@stanford.edu
Ren Hao: renhao@stanford.edu (not enrolled in the class)

Scammer actions	***Price & volume variations***
Scammers incrementally acquire long positions in these targeted stocks in a manner so as to avoid detection, i.e. no significant price movements	Consistent and significant increases in volume across a few days with minimal price impact
Scammers trigger sudden price increases through small but expensive buy orders, alongside an aggressive stock promotion campaign; "pump"	Surge in prices despite limited volume increase

Model

Let $\vec{X}_{i,t}$ be the vector of variables representing the financial attributes of a security i at time t where $\vec{X}_{i,t} = \left[X_{i,t}^{\{1\}} \; X_{i,t}^{\{2\}} \; X_{i,t}^{\{3\}} \; X_{i,t}^{\{4\}} \right] = \left[price_{i,t} \;\; trad_{volume_{i,t}} \; num_{shares_{i,t}} \;\; earnings_{TTM\,i,t} \right]$. Since most "pump-and-dump" stocks have a substantially lower price, we limit the our training examples to $i \in \{stocks\ s\ listed\ in\ NYSE/\ NASDAQ \mid X_{s,\tau}^{\{1\}} \leq 10\,, \tau = t(now)\}$ and $t \in \mathbb{Z}^+$ where $t = t(X) = number\ of\ trading\ days\ from\ 01Feb08\ to\ X$

Define $\vec{Z}_{i,t}$ as a vector of covariates capturing the past 20-days financial attributes of a stock i before time t such that
$\vec{Z}_{i,t} = \left[X_{i,t-20}^{\{1\}} X_{i,t-19}^{\{1\}} \ldots X_{i,t-1}^{\{1\}} \; X_{i,t}^{\{1\}} \;\; X_{i,t-20}^{\{2\}} X_{i,t-19}^{\{2\}} \ldots X_{i,t-1}^{\{2\}} \; X_{i,t}^{\{2\}} \;\; X_{i,t}^{\{3\}} \; X_{i,t}^{\{4\}} \right]$

$Y_{i,t}$ is the outcome indicator variable on forward-looking 5-day price movement such that $Y_{i,t} = \mathbb{I}(X_{i,t+5}^{\{1\}} \geq 1.5\, X_{i,t}^{\{1\}})$.

Data $\mathbb{X}$ would therefore be represented as a $|\, i \,| \, (|\, t \,| - 25) \times 42$ matrix

Denote $\hat{Y}_{i,t}$ as the model prediction for $Y_{i,t}$. Accordingly, we specify our loss function as
$\mathcal{L} = \sum Y_{i,t} \hat{Y}_{i,t} \left(\frac{X_{i,t+5}^{\{1\}} - X_{i,t}^{\{1\}}}{X_{i,t}^{\{1\}}} \right) - \alpha(1 - Y_{i,t}) \hat{Y}_{i,t}$.
The first term rewards the objective problem with each case of accurate and timely detection of "pump-and-dump" stock, weighted to the actual stock price appreciation. The second term penalizes false positives; $\alpha \in \mathbb{R}^+$ scales the penalty based on the investor's risk-tolerance profile as determined *a priori.*

In summary, we want our model's loss function to predict with extremely high likelihood whether a stock's price will increase by over 50% in the next 5 days, given the daily closing price and the volume of the stock for the past 20 days. We want our model to associate extremely high costs with false positives (where the model predicts that the stock's price will rise by 50% over the next 5 days but it does not happen). The loss function does this with the second term. The first term of the loss function simply lowers the cost for true positives.

Brandon Peh: bcp39@stanford.edu
Ren Hao: renhao@stanford.edu (not enrolled in the class)

Data Collection

Variable	***Data Extraction / Source***	***Data Processing***
$X_{i,t-20}^{\{1\}}\ X_{i,t-19}^{\{1\}}\ \dots\ X_{i,t-1}^{\{1\}}\ X_{i,t}^{\{1\}}$	Alpha Vantage (technical data)	Python
$X_{i,t-20}^{\{2\}}\ X_{i,t-19}^{\{2\}}\ \dots\ X_{i,t-1}^{\{2\}}\ X_{i,t}^{\{2\}}$	Alpha Vantage (technical data)	Python
$X_{i,t}^{\{3\}}$	Morningstar (fundamental data)	Python
$X_{i,t}^{\{4\}}$	Morningstar (fundamental data)	Python

Next Steps

Since the dataset has already been compiled, the next steps we are planning to undertake includes finding a suitable model that can utilize our data efficiently. Our current model uses the daily prices and volumes of past 20 days to give a binary prediction of whether the stock price will increase by 50% over the next 5 days. We plan to meet with our mentor to discuss possible models to implement since we do not have much expertise in the area of stock price prediction. Initial research points us towards the direction of building an RNN model with LSTM cells in tensorflow to predict the outcome.

However, given that we have changed the scope of our project towards the idea of penny stocks, we are still looking into alternative ideas and have yet to finalize our model. As a result, the baseline model has not been coded up yet as the idea was adapted into a different scope from what we had proposed in the project proposal, but we have included the code that helped us to collect and process the data (which is as essential as the model itself in contributing to the success of this project). As for splitting the dataset into training, test and dev sets, we are considering using past history as our training data set and the most recent 20% of our dataset as dev and test sets (10% each). By the end of week 8, we would be able to test a few models on our dataset and determine the best model to work with. By the end of week 9, we would probably have a fully trained model with fairly accurate results after numerous error analyses and tuning of hyperparameters. In week 10, we will begin working on the poster and final report in preparation for the final presentation. During the finals, the project would be completed!

Work Distribution:

Brandon: Collecting data and preprocessing the dataset
Ren Hao (not enrolled in the class): Brainstorming on models to implement

Brandon Peh: bcp39@stanford.edu
Ren Hao: renhao@stanford.edu (not enrolled in the class)

References:

1. https://pdfs.semanticscholar.org/0d21/aa97f8918f7384714131c3f3ea2a3abeb757.pdf
2. https://deeplearning4j.org/lstm.html
3. https://lilianweng.github.io/lil-log/2017/07/08/predict-stock-prices-using-RNN-part-1.html

Link for Github:

I am currently using a private repository, so I have added "cs230-stanford" as a collaborator as requested on piazza. My github account is bcp39@cornell.edu