Stock Movement Price Prediction with Behavioral Features

Research Thesis

In Partial Fulfillment of The Requirements for the Degree of Master of Science in Data Science

Rania Khoury

Submitted to the Senate of the Technion - Israel Institute of Technology

Tevet 5783, Haifa, December 2022

The Research Thesis Was Done in the Faculty of Data and Decision Sciences Under the Supervision of Dr. Ori Plonsky and Prof. Margarita Osadchy.

The author of this thesis states that the research, including the collection, processing and presentation of data, addressing and comparing to previous research, etc., was done entirely in an honest way, as expected from scientific research that is conducted according to the ethical standards of the academic world. Also, reporting the research and its results in this thesis was done in an honest and complete manner, according to the same standards.

The generous financial support of the Technion is gratefully acknowledged.

Table of Contents

Abstract

Predicting human behavior has long been a challenge for researchers, and behavioral economics has demonstrated the importance of behavioral biases in decision-making processes. Machine learning models have gained popularity in various fields for their ability to learn complex tasks. Recent studies have shown that incorporating behavioral insights into machine learning models can improve prediction accuracy. This study aims to expand upon previous research that demonstrated the benefits of incorporating behavioral insights into a stock market prediction model. Based on behavioral theories of decisions from experience, we assumed retail investors' trading patterns reflect sensitivity to the average return and to the probability of regret from a stock. We then hypothesized that because different groups of investors experience different average returns and probabilities of regret, it is possible to boost the predictive accuracy of a deep neural network developed to predict stock movements by adding features based on estimations of these two quantities for different groups of investors. We tested this hypothesis using both real-world stock data and synthetic stock data. The results suggest that when the signal behavioral features carry in prediction of stock prices is sufficiently strong accounting for these features can be useful. Yet, the results also indicate that real-world stock prices are too noisy for the models to provide useful predictions.

1. Introduction

1.1. Introduction

In recent years, we have been witnessing an increasing need for predicting human behavior in many real-world applications. For example autonomous driving in the field of artificial intelligence systems, trying to predict the trajectory of other agents in the scene so the robot can navigate a vehicle safely. Another example is marketing in the field of predictive analytics, marketers can identify patterns that can help predict consumer behavior in advance maximizing companies' profits. Another domain in which human behavior may impact real world outcomes is the financial sector, especially stock predictions.

Stock market prediction has been a challenging problem that has caught the attention of many researchers. Computer scientists and economists have tried to build a stock prediction model, using diverse methods. These methods include: Fundamental Analysis, the evaluation of a stock's price based on its intrinsic value e.g price to earnings ratio (a measure of expected earnings)[30]; Technical Analysis, the statistical analysis of price movements based on past trends like Moving Averages[31]; Linear models like ARIMA [6]; Machine Learning models like Logistic Regression and SVM [7]; and, most recently, deep learning models including Feed Forward, GRU, RNN, and LSTM (see, e.g, a github repository that includes a collection of deep learning in stock market prediction papers [8]).
Some researchers have made the observation that investors' behavior affects the stock market and thus accounting for said behavior may help predict stock movements. In an attempt to prove that accounting for investors' behavior might help predictive models, they introduce into their models features based on sentiment analysis of social media tweets\post\comments about stocks (e.g [7] [9].)

Going beyond data driven analysis of investor sentiments, it is possible that incorporating theory-driven behavioral insights into predictive models may also improve performance. Indeed, incorporating behavioral insights into a machine learning model is showing promising results in terms of prediction accuracy. For example, Plonsky et al. [10] show that by integrating psychology and data science they were able to produce the best predictor for aggregate human choice behavior in choice between gambles. The hybrid model beats purely behavioral models as well as data driven models without behavioral features.
In more recent work, Akiva [11] demonstrates that incorporating behavioral features into a stock prediction model increases the accuracy of the model. In his work, he incorporated Plonsky et al.'s idea [10] of adding behavioral features to machine learning to an Attentive LSTM network with

Adversarial training that has previously shown state of the art level stock predictions (Feng et al. [1]).
In our work, we aim to extend on Akiva's results by exploring the ability of a more fine grained definition of the behavioral features to improve the prediction power of the network further.

1.2. The impact of retail investors on the market

Retail investors have an impact on the market. They can influence market sentiment, which sets the tone in the financial markets through trading activity and price movements.
Mobile trading apps give retail investors accessibility to the trading market, along with their emotional, short-term trading patterns that can cause quicker reactions to the market.

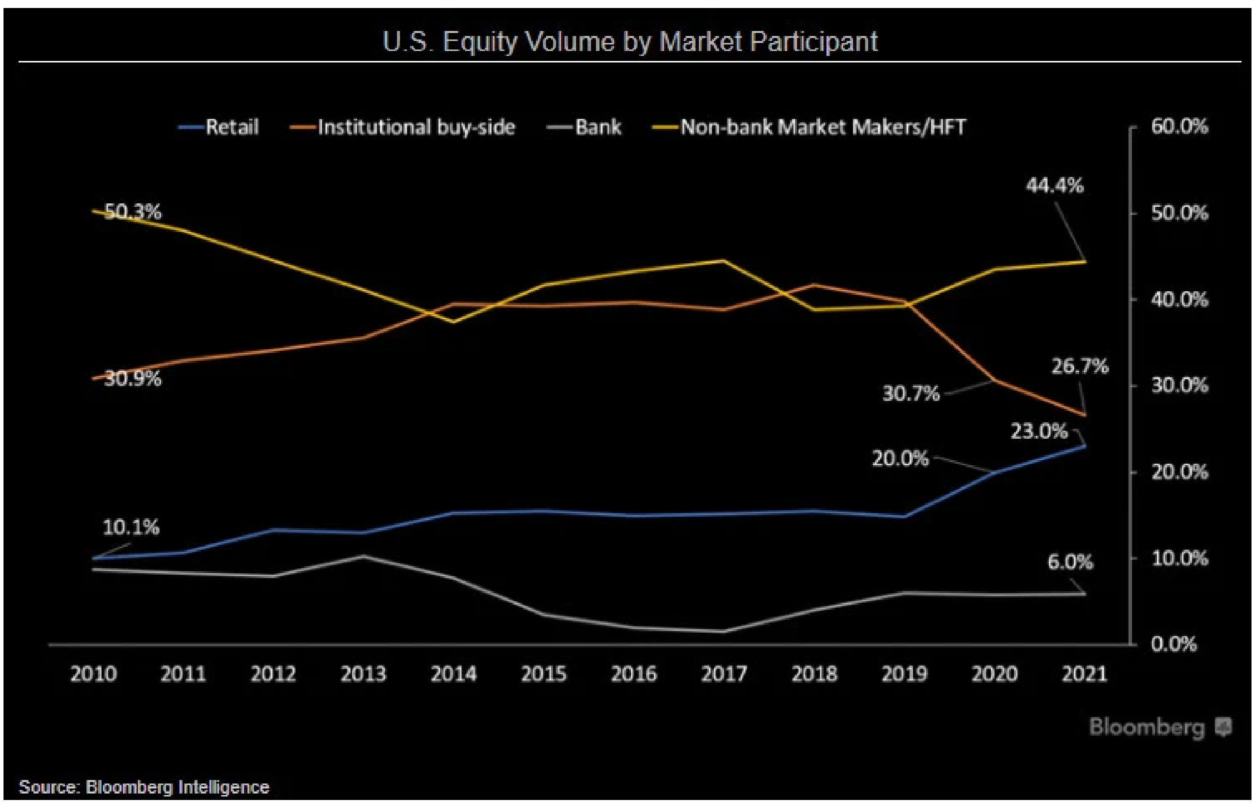

Figure 1: U.S. equity volume by market participant.
From "Stock-market gamification unlikely to end soon or draw new rules" . by Bloomberg Intelligence. February 19, 2021.
Copyright © 2022 Bloomberg Finance L.P.

Retail investors trade with their own personal finances, thus making them more emotionally invested in the market and are quick to react to earnings announcements or any other non-financial news positive or negative about the company regardless of the company's financial reports. As one anecdotal example, in May 2022 Tesla stock price dropped significantly following allegations of harassment against the company's CEO, despite the company's strong financial performance. This event serves as an example of how retail investors' perceptions of a company's future prospects can impact stock prices. In this instance, the allegations led many investors to lose confidence in the company's future and sell their shares, contributing to the drop in stock price and resulting in a loss of $10 billion in net worth for the CEO.

According to Bloomberg Intelligence (Figure 1), the market has been witnessing a sharp increase in retail investors' percentage since 2019. In 2021, the percentage was 23%, making them significant and impactful agents in the market.

Since retail investors impact the trading market, it is plausible that by predicting their behavior we can better predict the stock price movement. Unfortunately, direct prediction of investor behavior is a very difficult task. We do not have sufficient data about investors and their actions, nor do we know or can reliably estimate the mapping between investors' actions and stock movements. Instead, we can incorporate predictable behavior by retail investors in an already established stock price movement prediction model in order to enhance its accuracy. We will expand on our idea of incorporating retail investors' behavior in a prediction model through the remainder of this thesis.

1.3. Decisions from experience

There are several ways to predict retail investors' behaviors. For example, we can develop a machine learning model that predicts if the investor is going to sell/buy/hold. Yet because the relationship between behavior and stock price movements is hard to disentangle, we would like to use the power of a machine learning model to predict stock price movement directly. Integration of theory under this framework will be done using behavioral features. Plonsky et al. [10] show that by integrating behavioral features, based on psychological theories, in a machine learning system they were able to produce the best predictor for aggregate human choice behavior in choice between gambles. Therefore, incorporating theory-driven behavioral features into predictive models might be able to improve performance in the current setting as well.

To define our behavioral features, we borrow from the literature on decisions from experience (reviewed in [12]). In such tasks, the agent relies on past experiences that are similar to the current state and gains relevant experience in order to make a decision. Erev and Haruvy [12] argue that economic incentives determine the agents' experience, and this experience in turn derives future behavior. That is, the economic environment shapes decisions because it determines the decision makers' relevant experience. Often, studies of decisions from experience involve repeated choice between options with unknown underlying payoff distributions, but with feedback on previous choices that can shape future decisions. The stock market is a public context similar to the artificial environment used in such basic studies of decisions from experience: investors need to repeatedly decide between investments in stocks and get full feedback on their decisions (stock prices are a public information). In turn, this feedback generates the economic incentives that may shape future behavior. Hence, relying on findings from studies of decisions from experience to help predict stock movement is only natural.

Despite the complexity of choice behavior in general, basic studies of decisions from experience strongly suggest that a good basic model for prediction of repeated tasks has to account for two quantities of past experience that people are very sensitive to:

A. The average payoff obtained so far from each option.
B. The average of a small sample of past experiences taken from each option.

Indeed, in four different choice prediction competitions ([32],[33],[34] and [35]), where participants were asked to submit a model that predicts decisions from experience, the winning models demonstrated high sensitivity to the above-mentioned two quantities. Moreover, in a review of major drivers of people's decisions from experience, Erev and Roth [15] concluded that people demonstrate sensitivity to the average payoff generated by an option in the past (with similar sittings). In addition they argue: "experience leads to high maximization rates when the optimal choice also provides the best payoff most of the time (under similar settings), but not when this condition is not met." (p. 10818). This aspect of behavior can be captured by the idea that people rely on small samples of past experiences. Importantly, reliance on small samples implies high sensitivity to the probability of regret. That is, according to Erev and Roth, agents tend to choose alternatives that minimize the probability of regret.

It is worth mentioning that basic studies of decisions from experience lack some context that exists in the real world. Basic studies of decisions from experience usually involve simple laboratory tasks or games, such as choosing between two options with different probabilities of reward or punishment. Researchers' aim is to examine how people make decisions based on their past experiences without the influence of external factors such as market trends. In contrast, the stock market is a complex real-world environment involving many hard to control variables. The stock market involves millions of investors making decisions based on a wide range of factors, such as company's earning reports, news events, trends.
Another fundamental difference between these two environments is the time frame over which decisions are made. In basic studies of decisions from experience, participants often make decisions over a short time frame, such as a few minutes or hours; to further clarify, decisions in these contexts are often taken one after the other within seconds. While in the stock market, investors make decisions over much longer time frames such as days, weeks or months.
This lack of context may appear to be a limitation of the apparent applicability of the basic decisions from experience research to real world predictions. Yet, past research has also shown that these basic insights can

be applied in practice. Many important economic phenomena are the consequence of small decisions. For example Taleb's [37] prediction of the 2008 financial crisis. Taleb used the underweighting of rare events in decisions from experience to justify his "black swan" assertion, according to which investors tend to dismiss low-probability events. Another example is the many investors' tendency to have under-diversified investment portfolios (e.g Ben Zion et al. [38]), this tendency can be observed in the clicking paradigm and can be associated with agents' tendency to rely on past experience. Another example from real-world application that is not in the finance sector is the negative effect of punishments (Skinner [39]). In his work, Skinner argued that a student can learn to avoid punishments for bad academic performance by not coming to school. Skinner's observation was among the most important triggers for policies that banned the use of corporal punishments in school. Note that Skinner's observation builds on decisions from experience insights. The tendency to avoid punishment by dropping out of school can be a reflection of insufficient sensitivity to delayed outcomes. The teachers' tendency to punish bad performance can be a reflection of underweighting of rare events (that can be the result of reliance on small samples). From the teacher's point of view, the common consequence of punishment tends to be positive (the students try harder), and the problematic avoidance reaction is rare.

In the next section, we will discuss the design of two behavioral features that are based on the above-mentioned two quantities, and that can be used as input to a machine learning model.

1.4. Behavioral features in stock movement prediction

In Previous work, Akiva incorporated Plonsky et al. 's idea of adding behavioral features into a stock movement prediction model [11]. Specifically, to an Attentive LSTM network with adversarial training (Feng et al. [1]). He introduced two behavioral features that are consistent with the literature on decisions from experience:

- Grand Mean (GM): investors mean payoff from stock j in the previous t-1 days.

- Proportion of positive return (PR): the proportion of days with positive return from stock j in the previous t-1 days. (note this measure is dependant on the sign of the return and not its magnitude)

The GM feature captures the idea that when making repeated decisions, people tend to be sensitive to the average payoff an alternative provided in the past. In addition to being consistent with the literature on decisions from experience, the GM feature should be particularly useful for stock prediction since it captures people's known tendency to "chase" past returns in the market (e.g. Kliger et al. [13]). Specifically, people are sensitive to the observed average payoff of the stock in the past. This is explained by investor extrapolation, the belief that future expected returns are positively related to past stock returns (Barberis et al. [14]), which causes investors to select the option that caused the most desirable outcome in the past.

The feature PR captures the behavioral bias discussed in the previous section in which an agent tends to choose the alternative that minimizes the probability of regret. Regret in this sense is defined with respect to the alternative of holding cash. Hence the proportion of instances in which the observed stock return was positive can be considered an estimate for one minus the probability of regret.

The logic behind Akiva's work is that if indeed investors, like people in basic decisions from experience research, are mainly sensitive to the grand mean (GM) of the payoff from an option (stock) and to the probability of regret (PR) from that option over the previous t-1 days, then supplying these two features to a DNN should help it predict stock movements. Notably, even if (some) retail investors are indeed sensitive to these behavioral features (biases), it is difficult to know how they integrate it into their trading behavior. Yet the working assumption here is that there exists a function that maps this information to their trading behavior and this in turn impacts the stock price movements. Theoretically, a DNN should be able to approximate this function and therefore supplying these features to the DNN can be useful.

However, to compute the above-mentioned features for the previous t-1 days, it is necessary to know the window size (i.e. what t-1 is). In artificial decisions from experience tasks, the window size for the average payoff and probability of regret has a natural starting point: the beginning of the experiment. In the real world, this starting point should somehow be related then to the point at which the investor starts attending to the stock. A reasonable working assumption is that the window to which (some of the) investors are sensitive can be approximated by the period in which they are holding the stock. Admittedly, investors who are not holding the stock but are passively observing the stock's performance before making the decision of purchasing it for the first time may or may not be also impacted by GM and PR. However, data on such investors is not available. In the rest of this work, we assume that investors are likely to be sensitive to GM and PR starting from the moment they first start holding the stock and that this sensitivity has an effect on their trading behavior.

Importantly, different investors hold the stock for different periods, and this can lead to very different values for the above-mentioned features. For example, PR can be very different when investors hold a stock for 1 week or 1 year. In his work, Akiva neglected this complexity and assumed instead that useful features can be derived by replacing all investors with a "representative investor" with holding time that equals the average holding period of each stock over all investors.

To clarify why this may be an oversimplification, consider Figure 2 that shows how the probability of regret can differ among investors over time. The figure illustrates a situation in which there are 3 investors in a stock whose price over time is captured by the blue line. Investors 1 and 2 are weekly investors and investor 3 is a monthly investor.

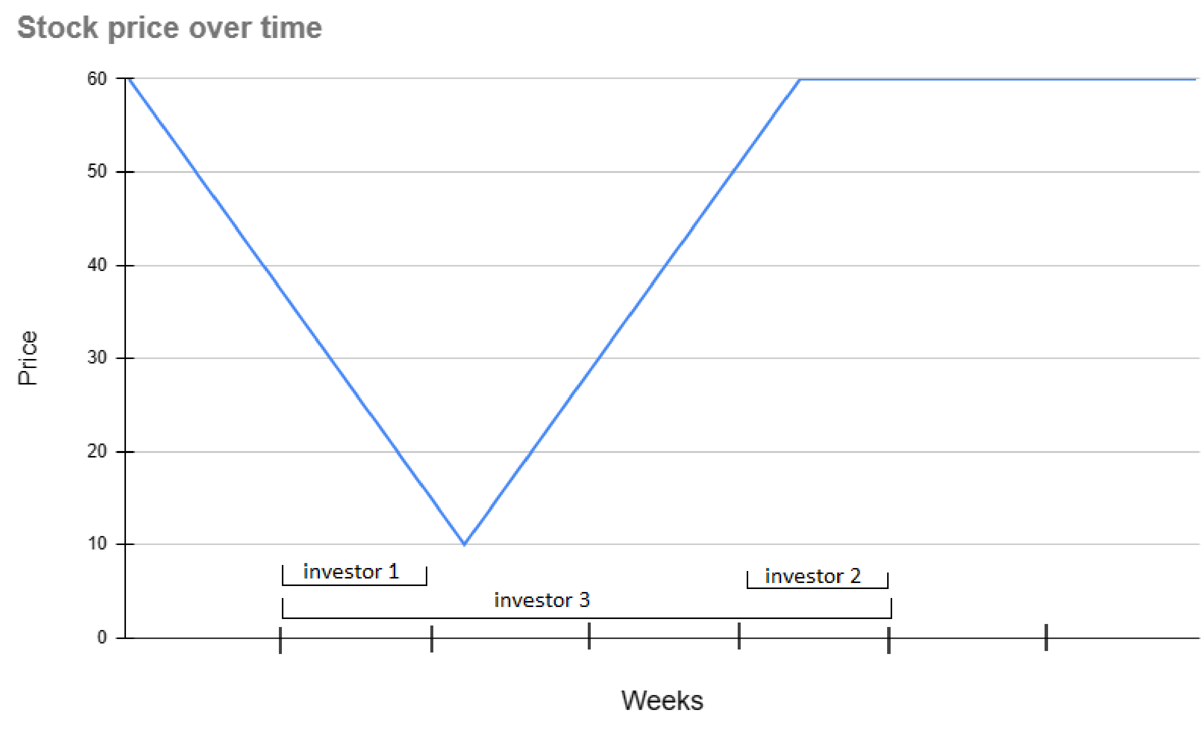

Figure 2: Illustration of how probability of regret differs for different types of retail investors

Without any calculations, we can see that investor 1 has a much higher probability of regret from holding the stock than investors 2 and 3 since in his investment window he only sees loss. While investor 2 has a higher probability of regret than investor 3, investor 3 sees a profit in his investment window and investor 2 sees no profits.Each of the 3 investors has a different probability of regret, which may lead to 3 different decisions in their stock investments, by which they can influence the stock market. (The same argument holds for the mean payoff).

This brings us to the following realization, Akiva's model has one clear limitation. Akiva uses the mean holding period over all investors to calculate PR and GM, but this assumption does not reflect the way by which different investors experience the stock market. Some investors are daily traders, some hold the stock for a week, some hold it for a month, and some for a year. In each given day, the probability of regret and the mean payoff for buying a specific stock for a weekly investor could be completely different from a monthly investor.

In this project, we want to introduce feature discrimination according to different groups of investors; each group would symbolize a different holding period length. For example, group ‘A’ has features corresponding to investors with an average holding period of 1 week, group ‘B’ has features corresponding to investors with an average holding period of 2 weeks, etc…

The Goal of this research is to expand on Akiva’s results to show that investors are sensitive to Probability of Regret and Grand Mean which is determined by their own holding periods, therefore using this information improves the accuracy of prediction models. Demonstrating the ability of this type of feature discrimination to improve upon previous results would lend support to the claim that investors are sensitive to this type of information (as would be predicted based on theories of decisions from experience) and that this sensitivity can be used to predict their real world behavior.

2. Stock movement prediction model

Our starting point for the development of a predictive model for stock movement is Feng et al’s Adversarial Attentive LSTM [1]. This model aims to predict whether the price of a stock will go up or down in the near future based on historical public data about the stock. The model’s novelty at the time of its publication was employing adversarial training to improve the generalization of the neural network prediction model. The model successfully became state of the art on two real-world stock data benchmarks outperforming the previous state of the art model (Xu and Chen [2]) with 3.11% relative improvement on average w.r.t accuracy.
The Attentive Adversarial LSTM mainly involves four components: feature mapping layer, LSTM, attention mechanism and adversarial training.

Feature mapping layer is a fully connected layer that is used to map the input data from one dimensional space into another, which can help the model to better capture the concealed relationships and patterns in the data. This can help the model to learn complex patterns which in turn improves its performance on the task. In addition, a feature mapping layer can help the model to generalize better to new data, as it allows the model to learn more abstract representations of the data that are not tied to specific input features. In the next subsections, we will discuss our changes to the architecture which are mainly changes to the input itself or an extension to the feature mapping layer.

LSTM (Long Short-Term Memory) is a type of recurrent neural network (RNN) that is particularly well-suited to sequential data. RNNs are a type of neural networks that are designed to process sequential data, such as time series, natural language. LSTMs are a variant of RNNs that have proven to be extremely effective on sequential data due to their ability to handle long-term dependencies in sequential data using memory cells and gates to control the information flowing through the network. In simpler words, LSTMs have the ability to store past important information and forget information deemed non important. This ability makes them compatible with time series prediction tasks where the ability of storing past information from the sequence is important.

Attention mechanism is a technique often used in RNN-based solutions for sequence prediction problems due to its ability to improve the performance of the model. The technique allows the model to focus on certain parts of an input sequence when processing it, rather than processing the entire sequence in a uniform way. This can be useful in sequence prediction tasks because it allows the model to attend to selective more important parts of the input for making a prediction. Attention mechanism can be considered a solution to a known challenge when working with sequential data, which is the need to process long sequences that may contain a lot of redundant information that may affect the model's prediction abilities. Simply put, the attention mechanism models the fact that data at different time-steps could contribute differently to the representation of the whole sequence. More particularly for stock representation, status at different time-steps might contribute differently.
A common approach to implement the attention mechanism is to use an attention layer that is trained to learn the weights of different parts of the

input sequence when making a prediction. The weights are later used to weight the contributions of the different parts of the input when generating a prediction.

The abilities of LSTM and the attention mechanism in sequence prediction tasks contribute to their popularity among researchers trying to predict the stock market. In a recent review, Jiang W. [8], documented over 31 publications in a span of 3 years that leveraged LSTM in their architecture. Specific examples include Qin et al. [4] study, in which they propose a dual-stage attention based recurrent neural network, which consists of an encoder LSTM unit with an input attention mechanism to adaptively extract relevant input features, and a decoder LSTM unit with a temporal attention mechanism to capture the long-range temporal information of the encoded inputs. Another noteworthy paper is Zhou et al. [36] where they propose a generic GAN framework employing LSTM and CNN for adversarial training to predict high frequency stock market.

Adversarial training is the key novelty in Feng et al's paper. It is used to improve the generalization of a neural network prediction model. The logic behind adversarial training in his architecture is that input features to stock prediction models are generally based on stock price, which is by its very nature a stochastic variable continuously changing as a function of time. Accordingly, normal training with static price-based features might easily overfit the data. To address the issue, Feng et al proposed to add perturbations to simulate the stochasticity of the price variable, and train the model to perform well under small intentional perturbations.

2.1. Investor Categorization

As explained above, this research is expanding on Akiva's results to show that investors are sensitive to Probability of Regret and Grand Mean which is a function of their own holding period. Investors hold their stocks for different periods of time. In each given day, the probability of regret and the mean payoff for buying a specific stock for a weekly investor could be

completely different from a monthly investor. Therefore, using this information might improve the accuracy of prediction models.
Individual investors' holding periods are not publicly available data. We are informed by a unique proprietary dataset on 379,882 users of an investment platform over a limited period of 6 weeks starting January 2019. We used this information to calculate the distribution of holding periods for each stock in our dataset for the holding periods (Categories) of 1 week, 2 weeks, 3 weeks, 4 weeks, 5 weeks, 6 weeks.
When Category 'K' weeks for stock 'S' is the proportion of investors who hold stock 'S' for more than 'K'-1 weeks and less or equal to 'K' weeks.
Note that investors for whom we have no complete information on holding period (e.g. holding the stock for more than 6 weeks) are excluded from the analysis.

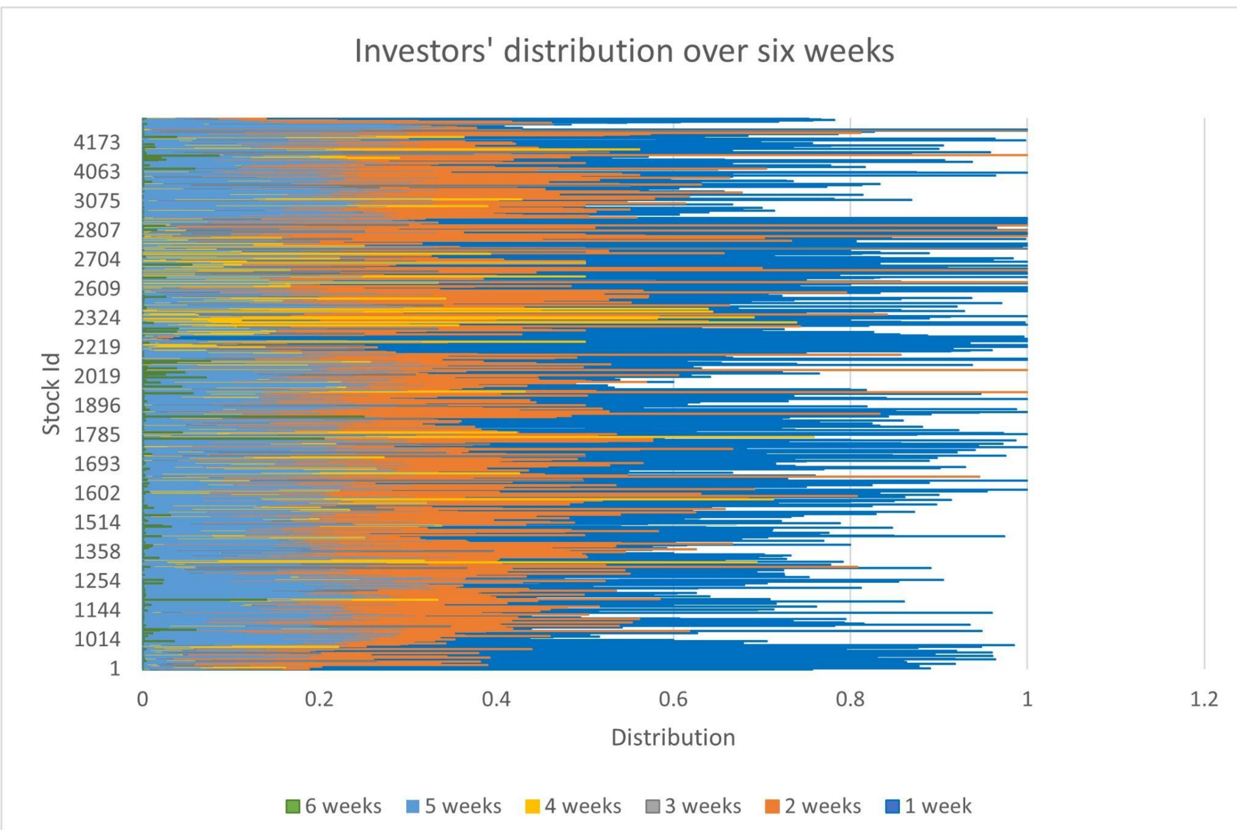

Figure 3: Illustration of retail investors' distribution over six weeks starting January 2019 in the private platform. Y-axis is a stock id, an id assigned to each stock symbol in our dataset.

According to Figure 3, which illustrates the distribution of investors on the private platform, it is uncommon for investors to invest in stocks for six weeks. Instead, the most popular investment periods among our investors are one, two and five weeks. With an average investment proportion of 0.543995511 for one week, 0.254723272 for two weeks, 0.031613848 for three weeks, 0.028964343 for four weeks, 0.134508999 for five weeks and 0.001669395 for six weeks. The figure below displays the average proportions of investors across the different categories along with their corresponding standard deviations.

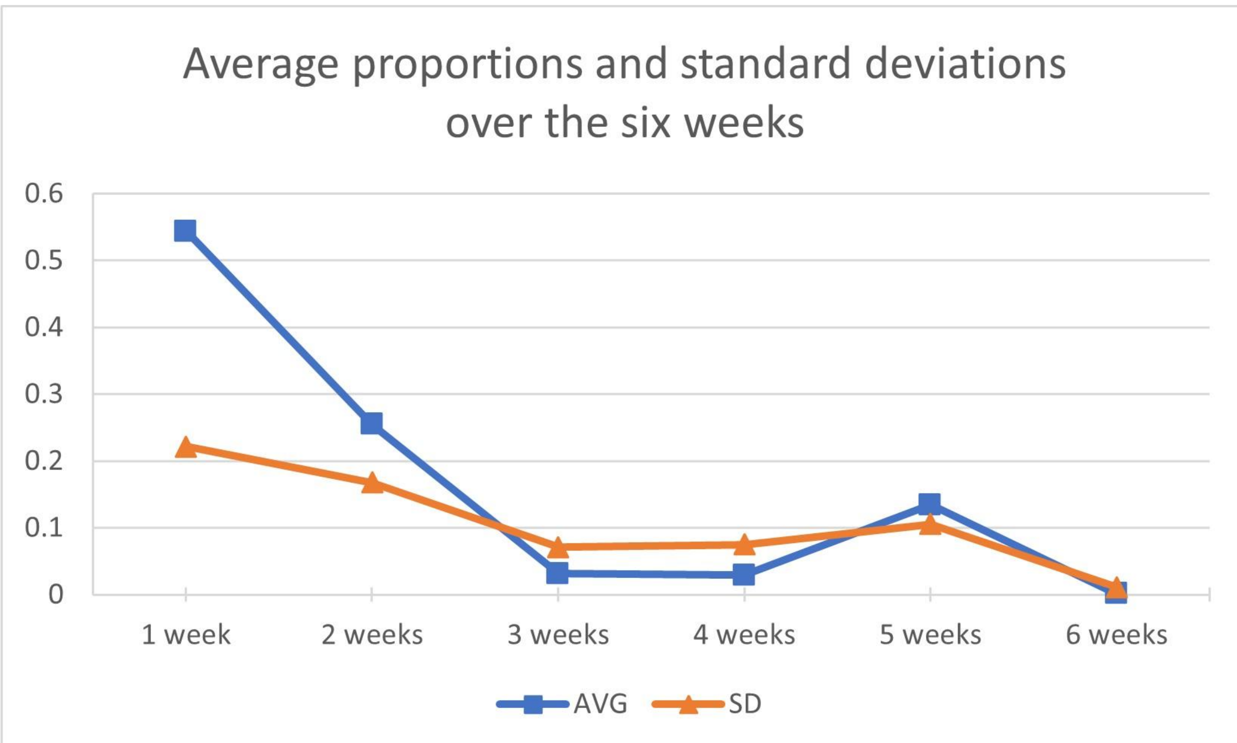

Figure 4: Illustration of the average proportion of investors across the six categories in our data and the standard deviations

Since we are limited by the available data, we assume that the distribution is applicable for users outside of the platform and in different time periods as well. This distribution was later used in features creation (explained in detail in *Experiments* section)

2.2. Features

Feng et al. [1] defined 11 technical features x^s_t to describe the trend of stock 's' at trading day 't':

Features	Calculation
c_open, c_high, c_low	e.g c_open = $open_t \, / \, close_t \; - \; 1$
n_close, n_adj_close	e.g n_close = $close_t \, / \, close_{t-1} - 1$
5-day, 10-day, 15-day, 20-day, 25-day, 30-day	e.g 5-day = $\frac{\sum_{i=0}^{4} adj\_close_{t-i} \, / \, 5}{adj\_close_t} - 1$

Table 1: Features to describe the daily trend of a stock in Feng et al.'s paper

$open_t$ is the opening price of a stock at day t.

adj_close_t is the adjusted price of a stock at day t. Which is the value of a stock after accounting for corporate actions such as stock splits, dividends, or other events that can affect the price of a stock.

The purposes of defining these technical features are to normalize the prices of different stocks (e.g., n_close) and explicitly capture the interaction of different prices (e.g., open and close).

On top of the above features, we added 12 features to capture investor's sensitivity to the average return and the probability of regret of a stock. We assume that investors are sensitive to the grand mean and the probability of regret from the moment they start holding the stock until they no longer hold it. Therefore, computing the value of these features based on the investors' holding time should be a logical consequence. We divided the investors into 6 categories corresponding to the 6 categories mentioned above, then we defined and calculated our features for each holding period category.

2.2.1. Features: Grand Mean (GM)

Grand mean is the observed mean return from stock 's' in the previous 'w' weeks.

$GM^{s}_{t,1}$ = grand mean for stock 's' on day 't' for a holding period of 1 week or less.
And in general:
$GM^{s}_{t,w}$ grand mean for stock 's' on day 't' for a holding period greater than 'w'-1 weeks and less or equal to 'w' weeks , $1 \leq w \leq 6$.
Meaning, we have 6 features of type grand mean.

$$GM^{s}_{t,w} = \frac{\sum_{i=0}^{7*w-1} adj\_close_{s,t-i} / 7*w}{adj\_close_{s,t}} - 1$$

2.2.2. Features: Proportion of Positive Returns (PR)

The proportion of days with positive return from stock 's' in the previous 'w' weeks; dependent on the sign of the return and not the magnitude.
This feature captures the minimization of regret. The proportion of days with positive return in a particular window of time is compared to holding cash which represents a payoff of zero over the same period of time.
An investor will choose an investment option that minimizes their probability of regret.

$PR^{s}_{t,1}$ = the proportion of positive return for stock 's' on day 't' for a holding period of 1 week or less.
And in general:

$PR^{s}_{t,w}$ = the proportion of positive return for stock 's' on day 't' for a holding period greater than 'w'-1 weeks and less or equal to 'w' weeks.

$$PR^{s}_{t,w} = \frac{\sum_{i=0}^{7*w-1} max(0, sign(\frac{adj\_close_{s,t-i}}{adj\_close_{s,t-i-1}} - 1))}{7*w}$$

2.2.3. Behavioral features integration

As explained before, even if some retail investors are sensitive to our behavioral features, we do not know how they integrate it into their trading behavior. Yet we assume that there exists a function that maps this information to their trading behavior which affects the stock price movements. Hence, incorporating these features to our machine learning model could help improve accuracy

2.3. Architecture

Incorporating behavioral features into Adversarial-LSTM Architecture can be accomplished in different approaches, we experimented mainly with two approaches with several tweaks for each approach.
In our early experiments, we left the building blocks of the architecture as is and combined the behavioral features along with the technical features in the input layer. Consequently modifying the original architecture by adding input vectors based on behavioral features.
In other experiments, in addition to combining the behavioral features along with the technical features in the input layer, we added embedding and concatenation layers. We divided our new input feature into 3 groups, and each group passed through their own embedding layer. Further discussion in our next two sections.

2.3.1. Architecture 1

The Attentive LSTM (ALSTM) mainly contains four components: feature mapping layer, LSTM layer, temporal attention, and prediction layer.

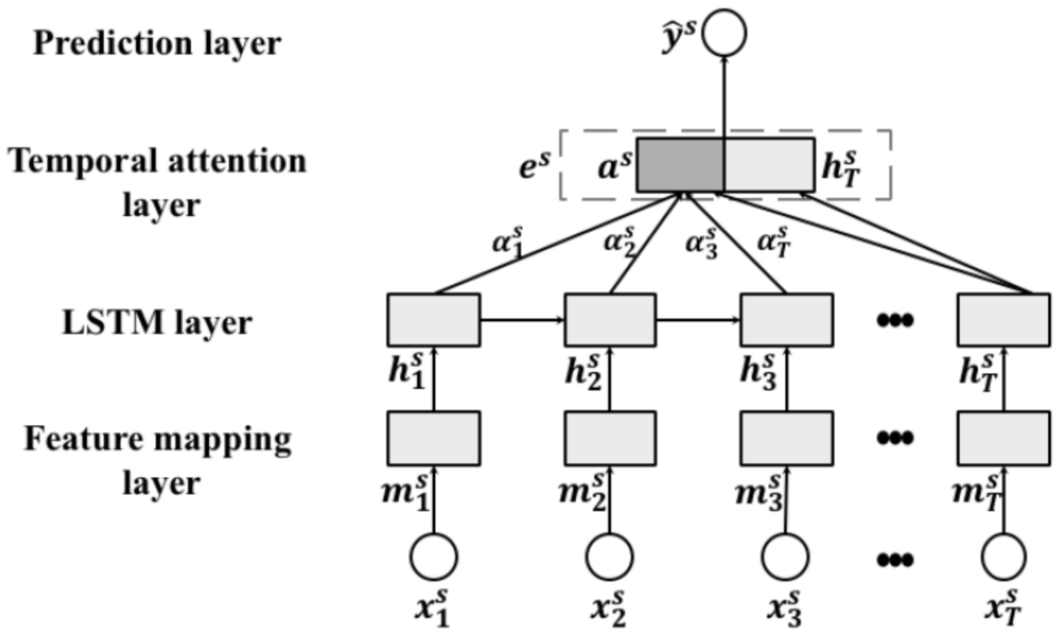

Figure 4: Illustration of the Attentive LSTM, source: Feng et al. [1].
T is a hyperparameter representing lag of length T explained in section 3.2

$x_t^s \in R^D$ $-$ is an input feature vector representing stock 's' status at day 't'.

e.g. open and close prices, gm features, pr features.
The input feature vector will be discussed further in our Experiments Section.

Feature Mapping Layer:
Previous work (Graves et al. [21], Wu et al. [22]) successfully shows that a deeper input gate benefits the modeling of temporal structures of LSTM greatly. Therefore a fully connected layer is applied to project the input features into a latent representation.

At each time step it performs as $m_t^s = tanh(W_m x_t^s + b_m)$ which projects the input features to a latent space with dimensionality of E.

The parameters $W_m \in R^{E\times D}$ and $b_m \in R^E$ are to be learned.

LSTM Layer:
Due to its ability to capture long term dependency, LSTM has been a popular choice to process sequential data (Qin et al. [4], Chen et al. [23]).

LSTM recurrently projects the input sequence into a sequence of hidden representations. At each time step, the LSTM learns the hidden representation h_t^s by jointly considering the input m_t^s and previously hidden representation h^s_{t-1} to capture sequential dependency.
We formulate the layer as $h_t^s = LSTM(m_t^s, h^s_{t-1})$. A more detailed formulation can be referred to (Hochreiter and Schmidhuber [25]).
An LSTM layer is applied to map $[m^s_1, ..., m^s_T]$ into hidden representations $[h^s_1, ..., h^s_T] \in \mathbb{R}^{U\,x\,T}$ with dimension of U to capture the sequential dependencies and temporal patterns in the historical stock features.

Temporal Attention Layer:
The attention mechanism has been widely used in LSTM-based solutions for sequential learning problems (Cho et al. [24], Chen et al. [23]). Attention is used to compress the hidden representations at different time-steps into an overall representation with adaptive weights in order to capture the fact that data at different time steps could contribute differently to the representation of the whole sequence.
Status at different time steps might contribute differently to stock representation. For example, days with maximum and minimum prices in the lag might have a higher contribution to the overall representation. Therefore, attention mechanism is used to aggregate the hidden representations as,

$$a^s = \sum_{t=1}^{T} \alpha_t^s h_t^s \text{ , } when\ \alpha_t^s = exp(\tilde{\alpha}_t^s) \,/\, \sum_{t=1}^{T} exp(\tilde{\alpha}_t^s)$$

$$\tilde{\alpha}_t^s = u_a^T tanh(W_a h_t^s + b_a)$$

The parameters $W_a \in R^{E' \times U}$ and $b_a, u_a \in R^{E'}$ are to be learned.

a^s is the aggregated representation that encodes the overall patterns in the sequence.

Prediction Layer:
According to (Fama and French [20]) the last hidden state is informative. Therefore instead of making prediction from a^s directly, we concatenate a^s with the last hidden state h^s_T, creating the final latent representation of stock s: $e^s = [{a^s}^T, {h^s_T}^T]^T \in R^{2U}$.
A fully connected layer is then used as the predictive function to estimate the classification confidence $\hat{y}^s = w_p^T e^s + b_p$.

The final prediction is $sign(\hat{y}^s)$.

2.3.2. Architecture 2

We added to the above architecture two components: embedding layer and concatenation layer.
Here, our input feature combines 11 technical features already defined by Feng et al. [1] with all 12 behavioral features;
$gm^s_{t,1}, \dots, gm^s_{t,6}, pr^s_{t,1}, \dots, pr^s_{t,6}$.

$x^s_t \in R^D$ – is an input feature vector representing stock ‘s’ status at day ‘t’, $D = 23$.

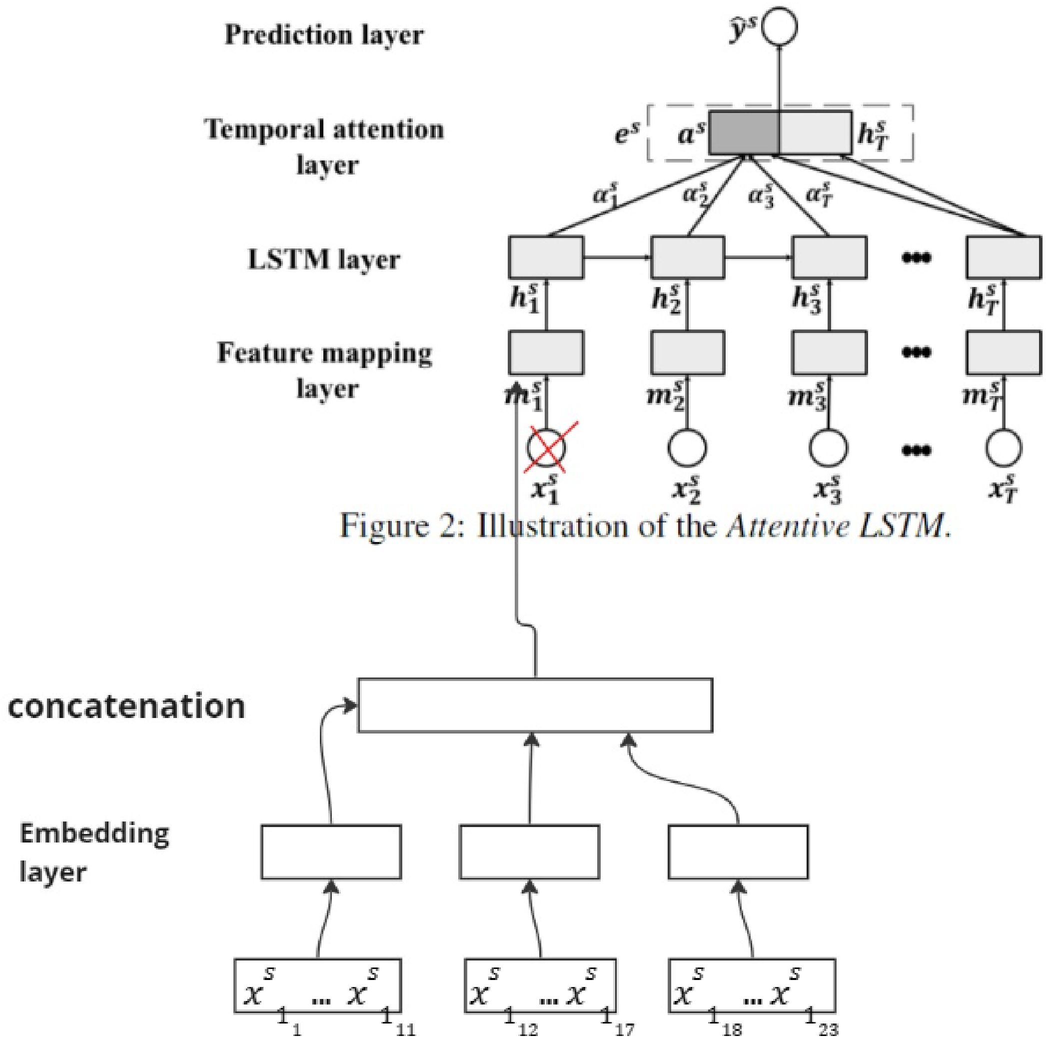

Figure 5: Illustration of the addition of embedding and concatenation layers on top of ALSTM

Embedding Layer:

Our input feature is a combination of two different categories: technical and behavioral. In a way, each category is coming from a different world.

Therefore, we divided our feature vector x_t^s into 3 smaller vectors:

$v^s_{t,1}$, $v^s_{t,2}$, $v^s_{t,3}$.

$v^s_{t,1}$ contains the first 11 features from x_t^s; the technical features defined by Feng et al. [1]

$v^{s}_{t,2}$ contains features 12 through 17; $gm^{s}_{t,1}, \ldots, gm^{s}_{t,6}$

$v^{s}_{t,3}$ contains features 18 through 23; $pr^{s}_{t,1}, \ldots, pr^{s}_{t,6}$

Each small vector passed through its own embedding layer.

$$e^{s}_{t,v_i} = tanh(W_e v^{s}_{t,i} + b_e) \in R^{E_i}$$

The parameters $W_e \in R^{E_i \times |v_{t,i}|}$ and $b_e \in R^{E_i}$ are to be learned.

Concatenation:
The result of all the embeddings is then concatenated together in the following manner: $m^{s}_{t} = concat(e^{s}_{t,v_1}, e^{s}_{t,v_2}, e^{s}_{t,v_3})$

m^{s}_{t} is then fed to Architecture 1 as the new input.

2.4. Adversarial training

To train a classification model, we need to minimize an objective function Γ. Usually hinge loss [16] is used in optimizing classification models.

$$\Gamma = \sum_{s=1}^{S} l(y^{s}, \hat{y}^{s}) + (\alpha/2)||\Theta||_F^2, \quad l(y^{s}, \hat{y}^{s}) = max(0,\ 1 - y^{s}\hat{y}^{s}).$$

The first term is hinge loss and the second term is a regularizer on the trainable parameters to prevent overfitting.

Stock price is a stochastic variable, continuously changed with time and is affected by stochastic trading behaviors at a particular time-step (Musgrave, 1997 [28]). Our features are calculated from stock prices which means normal training might lead the model to overfit the data and to lack generalization ability. This is because normal training assumes the input is static, which leads to the realization that normal training is inappropriate for stock price movement prediction models.

Therefore we use *adversarial training* (Goodfellow et al. [17], Kurakin et al. [18]) since it addresses the stochasticity of our data by training the model on both clean examples (i.e examples in the training set) and *adversarial examples* (AEs) (Szegedy et al. [19]).
Adversarial examples are malicious inputs generated by adding intentional perturbations to features of clean examples. The perturbation, named as *adversarial perturbation* (AP), is the direction that leads to the largest change of model prediction. By enforcing the predictions on the adversarial examples to be the same as their corresponding clean examples, the model could capture the stochasticity of stock prices.

Applying adversarial training on stock prediction has two challenges. First, calculating perturbations relies on calculation of gradients regarding the input; backpropagation through time step of the LSTM layer is time consuming. Second, gradients of the input are dependent across different time steps which might lead to unintentional interactions among the perturbations on different time steps, which are uncontrollable.
To address these challenges, Feng et al. [1] proposes to generate adversarial examples from latent representation e^s, as shown in figure 6.

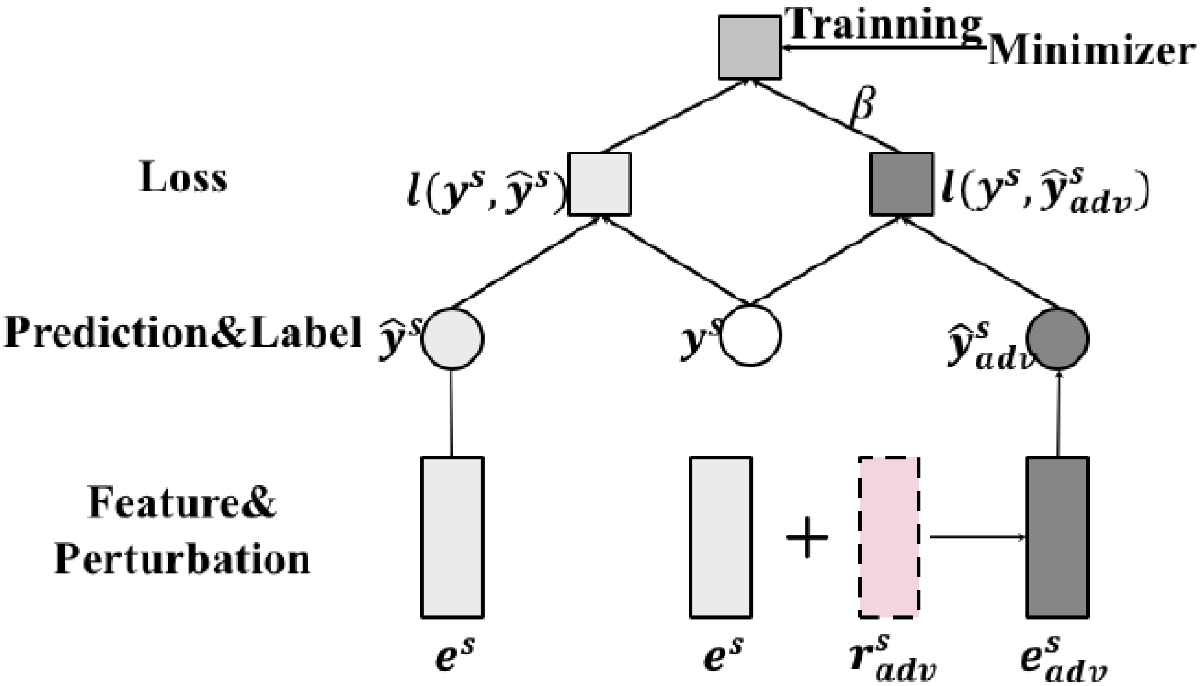

Figure 6 : Illustration of the Adversarial Attentive LSTM.
Image source: Feng et al. [1]

The objective function is then adapted to adversarial training.

$$\Gamma_{adv} = \sum_{s=1}^{S} l(y^s, \hat{y}^s) + \beta \sum_{s=1}^{S} l(y^s, \hat{y}^s_{adv}) + (\alpha/2)||\Theta||_F^2$$

the second term is an adversarial loss where $\hat{y}^s_{adv}$ is the classification confidence of the adversarial example of stock s. β is a hyperparameter to balance the two losses.
Minimizing the objective function encourages the model to correctly classify both clean and adversarial examples.
Since adversarial perturbation is the direction leading to the largest change of model prediction then by classifying an adversarial example correctly, the model can make correct predictions for examples with arbitrary perturbations at the same scale. Therefore, adversarial training could enable ALSTM to capture the stochasticity of stock inputs.
At each iteration, the latent representation of an adversarial example is generated according to the following formulation:

$$e^{s}_{adv} = e^{s} + r^{s}_{adv} \quad , r^{s}_{adv} = arg\ max_{r^{s}, \|r^{s}\| \leq \epsilon} l(y^{s}, \hat{y}^{s}_{adv}),$$

Where e^{s} is the final latent representation of stock s, ϵ is a hyperparameter to control the scale of perturbations.

r^{s}_{adv} is the direction (perturbation) that leads to the largest change of model prediction, Therefore can be calculated using the fast gradient approximation method (Goodfellow et al. [17]),

$$r^{s}_{adv} = \epsilon \frac{g^{s}}{\|g^{s}\|_2} \quad , \quad g^{s} = \frac{\partial l(y^{s}, \hat{y}^{s})}{\partial e^{s}},$$ under L_2 norm.

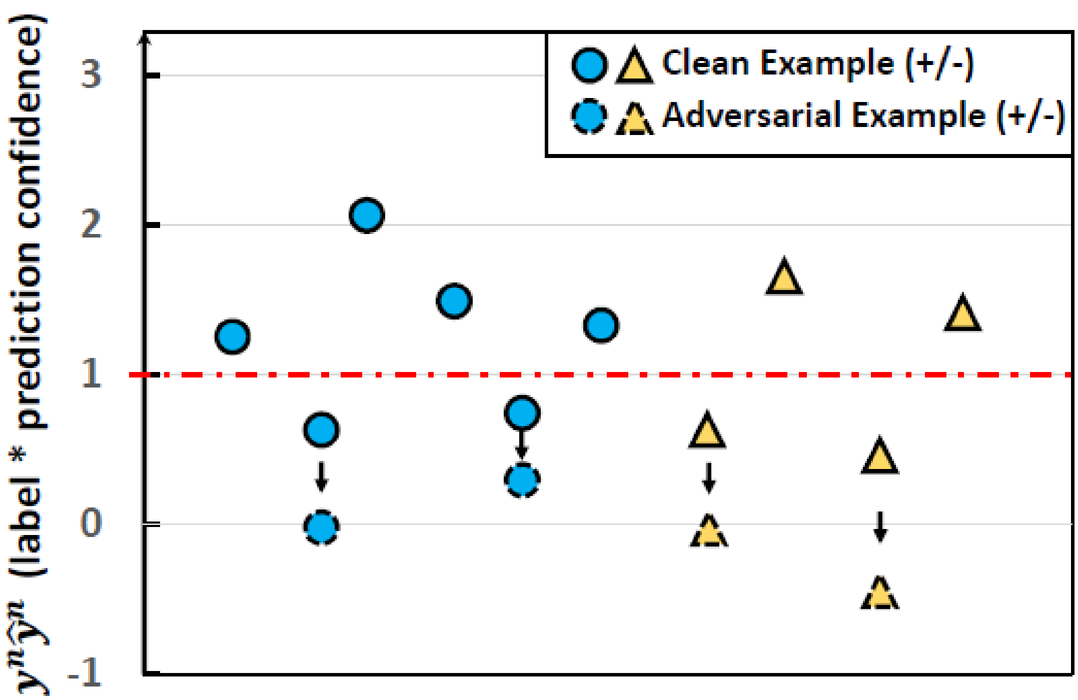

Figure 7: Intuitive illustration of adversarial examples.
Image source: Feng et al. [1]

In a training iteration, an adversarial example is generated when a clean example has a loss larger than 0 (i.e. $y^{s}\hat{y}^{s} < 1$). The model is then updated to minimize the loss for both clean and adversarial examples, forcing the margin between clean examples and the decision boundary.

Therefore, predicting adversarial examples into the same class as the clean examples benefits the model greatly.
Adversarial training encourages the model to correctly predict samples drawn from the inherent stochastic distribution of inputs, causing the model to capture stochasticity. Since the adversarial perturbation (r^s_{adv}) varies between iterations, the adversarial training adaptively adjusts the strength of enforcing margins during the training process, while hinge loss pushes the decision boundary away from examples close to the boundary.

3. Experiments

3.1. Preliminaries

Since Feng et al's architecture is the foundation of our work, first we had to make sure we can run, operate and understand all parts of Feng et al's code. Unfortunately the code was written in Tensorflow 1.8, an outdated version of Tensorflow which is abandoned by Academia and enterprises, and more crucially is no longer supported by our (new) current PCs due to new GPUs. In the interest of having a project that is relevant to modern technology, with most of the data scientists, researchers and enterprises transitioning to PyTorch, we converted the project's code into Pytorch.
After the conversion was done, we ran many tests on our new project on the same datasets used in Feng et al's [1] paper to make sure that our new project produces the same results documented in the paper, and that it behaves similar to the old code.
Only after we made sure that we can reproduce both Feng and Akiva's [11] reported results that we started our own experimentation.

3.2. Experimental Settings

We evaluated our methods on two benchmarks on stock movement prediction, ACL18 (Xu and Cohen [2]) and KDD17 (Zhang et al. [5])
ACL18 contains historical data from January 1st 2014 to January 1st 2016 of 88 high-trade volume stocks in NASDAQ and NYSE markets.

First, we aligned the trading days in history, i.e. removing weekends and public holidays that lack historical prices. Candidate examples were constructed by moving a lag of length T along the aligned trading days (i.e., one example for a stock on every trading day). The label of each example was determined according to the movement percent of stock close prices. Given a candidate example of stock s in the lag $[T' - T + 1, T']$, the movement percent is calculated as $p^{s}_{T'+1}/p^{s}_{T'} - 1$, where $p^{s}_{T'}$ is the adjusted close price of stock s on day T'. Following previous works, examples with movement percent $\geq 0.55\%$ and $\leq - 0.5\%$ are labeled as positive and negative examples respectively, ignoring minor movement ratio. The examples were split into training (January 1st 2014 to Aug 1st 2015), validation (Aug 1st 2015 to Oct 1st 2015) and testing (Oct 1st 2015 to January 1st 2016).

KDD17 contains historical data from January 1st 2007 to January 1st 2016 of 50 stocks in U.S. markets. The same procedure as ACL18 was followed to identify positive and negative examples. The constructed examples were split into training (Jan 1st 2007 to Jan 1st 2015), validation (Jan 1st 2015 to Jan 1st 2016) and testing (Jan 1st 2016 to Jan 1st 2017).

In addition, we have access to a unique proprietary dataset on 379,882 users of an investment platform over the period of 6 weeks starting January 2019. This data was used in order to calculate distributions of holding periods over the investors for each stock as explained in Section 2.1. Notably, our stock price training data ends in 2016 while our 6 week proprietary data is from early 2019. We nevertheless assumed that the elicited distribution of holding periods is applicable for different time periods.

3.3. Integration of behavioral features

The technical and behavioral features are explained in section 2.2.
In this section we are addressing the different approaches in which we combined our behavioral features along with the 11 technical features.

3.3.1. Approach 1 - uniformly distributed

12 behavioral features gm_{1week}, .., gm_{6weeks}, pr_{1week}, ..., pr_{6weeks} are added on top of the already defined 11 technical features.
We assumed that investors are uniformly distributed, meaning investors are equally likely to be in each category of the holding periods (1 week, 2 weeks, ... , 6 weeks).
The behavioral features were calculated for each day of our two datasets to construct an input vector $x^s_t \in \mathrm{R}^{23}$.

3.3.2. Approach 2 - $P_i F_i$

12 behavioral features gm_{1week}, .., gm_{6weeks}, pr_{1week}, ..., pr_{6weeks} are added on top of the already defined 11 technical features.
We assumed the investors distribution follows the distribution calculated from the 6 weeks data. The behavioral features were multiplied by their respective distribution.
For example, let's say we have a stock 'XX' and its investors distribution is:
{1 week - 0.75, 2 weeks - 0.204545, 3 weeks - 0.045455, 4 weeks - 0,
5 weeks - 0, 6 weeks - 0 }
Then $gm^{XX}_{t,1} = 0.75 \cdot gm^{XX}_{t,1}$, $pr^{XX}_{t,1} = 0.75 \cdot pr^{XX}_{t,1}$

$gm^{XX}_{t,6} = 0 \cdot gm^{XX}_{t,6}$, $pr^{XX}_{t,6} = 0 \cdot pr^{XX}_{t,6}$.

The behavioral features were calculated for each day of our two datasets to construct an input vector $x^s_t \in \mathrm{R}^{23}$.

3.3.3. Approach 3 - weighted behavioral features

Only 2 behavioral features $gm_{weighted}$, $pr_{weighted}$ are added on top of the already defined 11 technical features.
When each feature is the weighted average of its 6 behavioral features with respect to the calculated distribution.

$gm_{t,weighted} = \sum_{i=0}^{5} w_i \cdot gm_{t,i}$, when w_i is the proportion of investors holding the stock for longer than 'i-1' weeks and less or equal to 'i' weeks.

$\sum_{i=0}^{5} w_i = 1$ Since this is a distribution.

The behavioral features were calculated for each day of our two datasets to construct an input vector $x_t^s \in \mathbb{R}^{13}$.

3.4. Baselines

We compared our model to the following methods documented in Feng et al. [1]:

- Momentum (MOM): a technical indicator, predicts negative or positive for each sample according to the trend in the last 10 days

- Mean Reversion (MR): predicts the movement of each sample as the opposite direction of the latest price towards the 30-day moving average.

- Adv-ALSTM: Adversarial Attention LSTM (Feng et al. [1]), which is similar to ALSTM with adversarial training instead of normal training. The input is our 11 technical features in table 1.

- Adv-ALSTM-Psycho (Akiva's model): which is the same as Adv-ALSTM. The input is our 11 technical features with an addition of two behavioral features GM and PR, calculated based on the mean holding period (assuming a representative investor).

3.5. Evaluation

In this study, we aim to evaluate the contribution of incorporating behavioral features into the input of the network. To do this, we first train the “Base” network that is described in section 2.3.1 without any behavioral features. Using the accuracy on the validation set in each dataset to select the best architecture. Once the architecture has been fixed, we evaluate the Base model’s prediction performance on the test set using the two evaluation metrics listed below (accuracy and MCC). Using the same fixed architecture that was chosen to maximize predictive accuracy without behavioral features, we examine the predictive performance (on the test set) of variants of this architecture that include access to behavioral features (see sections 2.3 and 3.3). The results thus reflect the relative contribution of the addition of the behavioral features to an architecture that does not include them.

The prediction performance is evaluated using two metrics, Accuracy (Acc) and Matthews Correlation Coefficient (MCC) (Xu and Cohen [2], Feng et al. [1]). Accuracy is the proportion of the total number of predictions that were correct (true positives and true negatives), in the range [0, 100]. MCC is a reliable statistical rate which produces a high score only if the prediction obtained good results in all of the four confusion matrix categories (true positives (TP), false negatives (FN), true negatives (TN), and false positives (FP)), proportionally both to the size of positive elements and the size of negative elements in the dataset, defined as:

$$MCC = \frac{TN \cdot TP - FN \cdot FP}{\sqrt{(TP+FP)\cdot(TP+FN)\cdot(TN+FP)\cdot(TN+FN)}}$$

The range of MCC is [-1, 1], and 0 reflects the score of random guessing. Note that the higher the value, the better the performance for both metrics.

3.6. Parameter Settings

We implemented Adv-ALSTM and Adv-Psychological-ALSTM with pyTorch and optimized it using mini-batch Adam (Diederik and Jimmy [29]) with a batch size of 1,024 and learning rate of 0.001 for ACL18 and a learning rate of 0.01 for KDD17, and we tuned the rest of the hyperparameters, the number of hidden units (U), lag size (T), weight of regularization term (λ) and (β) on the validation set to optimize Adv-Psychological-ALSTM model. Tuning was done using grid search within the ranges [4, 8, 16, 32], [2, 3, 4, 5, 10, 15], [0.0001, 0.001, 0.003, 0.01, 0.1, 1] and [0.001, 0.005, 0.01, 0.05, 0.1] for U, T , λ and β respectively. Testing performance is reported when Adv-Psychological-ALSTM performs best on the validation set.

In addition to tuning the hyperparameters, we also experimented with shuffling the train and validation sets before constructing examples as explained in section 3.2. This was done to try and help the model to better generalize since the distribution of the historical data changes over time as is the nature of stock prices. Note, however, that all results are given for the test set which is not shuffled in any way.

3.7. Results and discussion (real world data)

The results of our analysis are shown in Table 2. These results show that the predictive accuracy of all models is not very different from chance, with MCC near zero in all cases. Importantly, this poor predictive performance is clear already in the Base model. This highlights the difficulty in predicting stock movements in general.

Shuffle	Model	ACL18		KDD17	
		Acc	MCC	Acc	MCC
	MOM	47.01	-0.064	49.67	-0.013
	MR	46.22	-0.078	48.46	-0.036
	Adv ALSTM Psycho	51.31	0.026	51.79	0.024
Yes	Base	50.22	0.005	51.50	0.021
	Uniformly distributed	50.37	0.008	51.71	0.011
	$P_i F_i$	50.67	0.011	51.80	0.011

	Weighted behavioral features	**51.74**	0.033	**52.36**	0.014
	Embeddings	51.08	0.022	52.18	0.021
No	Base	49.99	0.0005	51.93	0.032
	Uniformly distributed	49.67	-0.004	50.49	0.006
	P_iF_i	51.90	0.036	52.04	0.020
	Weighted behavioral features	51.00	0.019	**52.32**	0.027
	Embeddings	**51.89**	0.038	51.19	0.024

Table 2: Performance comparison on benchmarks ACL18 and KDD17 datasets
Adv-Psycho-ALSTM is previous work done by Ori AKica
Base - Architecture 1 without behavioral features
Embeddings - Integrating behavioral features using approach 1 (section 3.3.1) in architecture 2.

However, our main concern is in the relative improvements of models that include the behavioral features beyond the Base model. From this perspective, the results reveal the effectiveness of behavioral features in improving accuracy. While the increased accuracy in absolute percentage points is very small (under 2%), the relative improvement is more promising.

Specifically, our analysis indicates that the weighted behavioral features model improves the relative accuracy by 3% on the ACL18 benchmark when train and validation sets are shuffled during training. While the Embeddings model (architecture 2) relatively improves accuracy by 3.8% on the same benchmark when the train and validation data are not shuffled. Similarly, the weighted behavioral features model improves the relative accuracy by 1.6% when train and validation sets are shuffled and by 0.75% when they are not shuffled. The relative improvement in terms of MCC, especially on ACL18 are much larger, although this is mainly the result of a near zero MCC for Base in that setting.

4. Synthetic data

The results of the experiment suggest a negligible increase in absolute accuracy. However, the observed relative improvement in performance suggests the potential effectiveness of incorporating behavioral features in predicting stock price movement. This raises the question of whether the inclusion of such behavioral features can truly improve prediction accuracy, or are the current results of relative improvements an artifact of the very low base accuracy. Specifically, it is possible that in a setting in which the current architecture performs better than chance, the addition of behavioral features will not contribute beyond the signal captured by the technical features.
To address these possibilities, we generated two synthetic datasets of stock prices that are directly related to the behavioral features under investigation. By evaluating the ability of our models to predict stock movements in these synthetic data, we aim to determine if a significant improvement in prediction accuracy can be achieved with the inclusion of behavioral features, under the same architectures, and even when the base results are better than chance. This analysis will allow us to determine if behavioral features do indeed have the potential to contribute to an enhancement in model performance using the current framework.

4.1. Method

We simulate stock movements over a period of 9 years using the Geometric Brownian Motion model ([26][27]), a commonly used approach in financial research for this purpose. Our simulated close prices are a function of our 12 behavioral features, calculated day after day in chronological order (the explicit function will be discussed in further detail below), with the open price of each day equaling the close price of the previous day. ($open_t = close_{t-1}$, t indicates the number of the day). In an attempt to increase the signal of our behavioral features in the data.

After simulating the stock, we constructed examples following the same logic used for ACL18 and KDD17, and split them into training (December 9th 2013 to January 3rd 2022), validation (January 3rd 2022 to July 1st 2022) and testing (July 1st 2022 to December 7th 2022).
We trained and tested our simulated stock on Base and Behavioral models that follow Architecture 1

4.2. Geometric Brownian Motion model of stock movements

The most important object to model when modeling a stock market is the return. We can calculate the return of day 't' according to this formula:
$r_t = \frac{p_t - p_{t-1}}{p_{t-1}}$, when p_t is the price of the stock at day 't'.
Given the above formula, we can extrapolate the price time series using the following formula:

$$p_t = p_0 \prod_{i=0}^{t} (1 + r_i)$$

Hence, by knowing the starting price p_0, we can calculate future prices using the sequence of returns. In order to simulate a stochastic behavior of stock, we use the Geometric Brownian Motion model (GBM) (Sengupta [26], Ermogenous [27]), in which the returns are uncorrelated and normally distributed; $r_i \sim \mathcal{N}(\mu, \sigma)$.
μ is the mean value of the returns, if it's positive we have a bullish trend. If it's negative we have a bearish trend.
σ is the volatility of the returns. The higher this value compared to μ the more unstable the price.
GBM (Sengupta [26], Ermogenous [27]) is a particular model of the stock market often used in financial mathematics because it's the simplest stock market model one can build.

4.3. Synthetic history

To simulate stock movements based on behavioral features values, we first need to create a short synthetic 60-day history of a stock and that is because the feature values are calculated based on past stock prices.

We generate 60 normally distributed random variables and calculate future prices starting from a start price according to the GBM model with
$\mu = 0.002$, $\sigma = 30$ and start-price = 30.
We then calculate a history of 60 days resulting in the synthetic stock shown in Figure

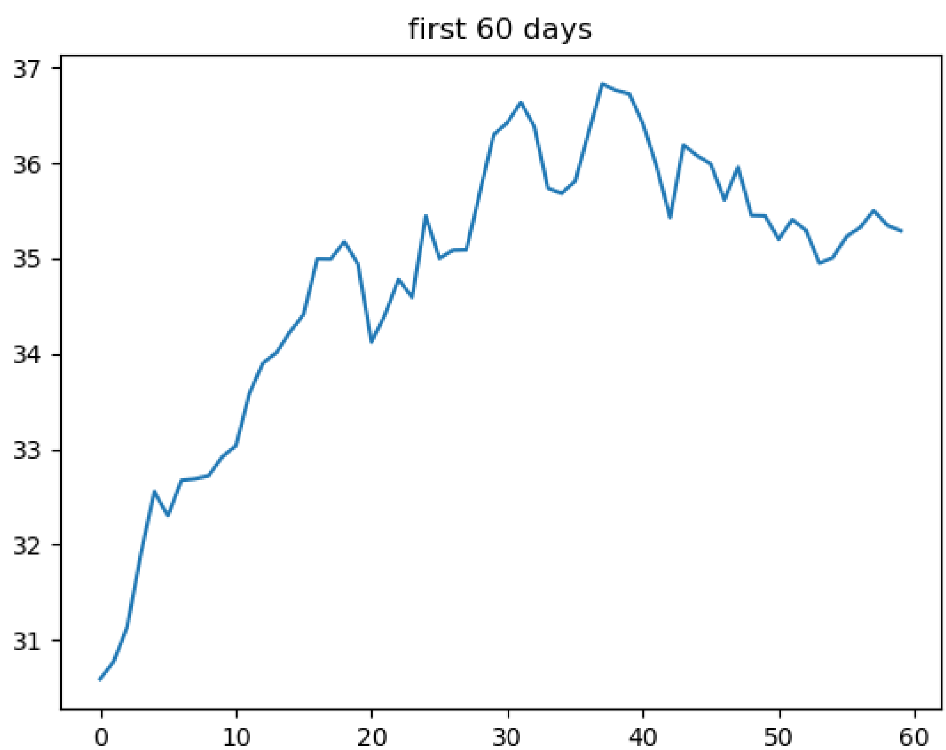

Figure 8: Synthetic stock history prices for the period of 60 days generated using GBM model. x-axis is days and y -axis is price

Now that we have a history of stock prices we can calculate our 12 previously discussed behavioral features for the 60th day.

4.4. Synthetic data 1

Our first synthetic data was created by using the following value for the mean of the price return in each day t:

$$\mu_t = \beta_1 \cdot \sum_{i=0}^{5} dist_i \cdot \sin(2\pi c_{i_1} \cdot gm_{t,i} + c_{i_2}) + \beta_2 \cdot \sum_{j=0}^{5} dist_j \cdot \sin(2\pi c_{j_1} \cdot pr_{t,j} + c_{j_2})$$

$$close_t = open_t + \mathcal{N}(\mu_t, \sigma_{const})$$

The following values were assigned to to our constants:

- $gm\ constants\ -\ c_{i_1}, c_{i_2}$

$c_{0_1} = 100,\ c_{0_2} = 0,\ c_{1_1} = 200,\ c_{1_2} = 2,\ c_{2_1} = 3,\ c_{2_2} = 3, c_{3_1} = 10,\ c_{3_2} = 4$
$c_{4_1} = 40,\ c_{4_2} = 5,\ c_{5_1} = 1,\ c_{5_2} = 6.$

- $pr\ constants\ -\ c_{j_1}, c_{j_2}$

$c_{0_1} = 300,\ c_{0_2} = 7,\ c_{1_1} = 5,\ c_{1_2} = 8, c_{2_1} = 20,\ c_{2_2} = 9,\ c_{3_1} = 30,\ c_{3_2} = 10$
$c_{4_1} = 60,\ c_{4_2} = 11,\ c_{5_1} = 6,\ c_{5_2} = 12.$

Causing a sum of Sin functions of different frequencies.
where $\beta_1 = 0.8$, $\beta_2 = 0.2$, $\sigma_{const} = 0.01$ are constants as well.
$dist_i$ is the normalized mean of all distributions of the stocks we have in ACL18 and KDD17. In doing so we tried to insert a distribution of investors affecting the stock movement to simulate the real world data. Hence, this synthetic data can be thought of as representing a weighted average of the effects of the different groups of investors on the trend of stock return. Each such effect is a (sinus) function of the group of investors' observed feature values. Hence, the data here represents a direct signal linking groups of investors' features with the stock movements.

The resulting simulated stock can be seen in Figure 9.

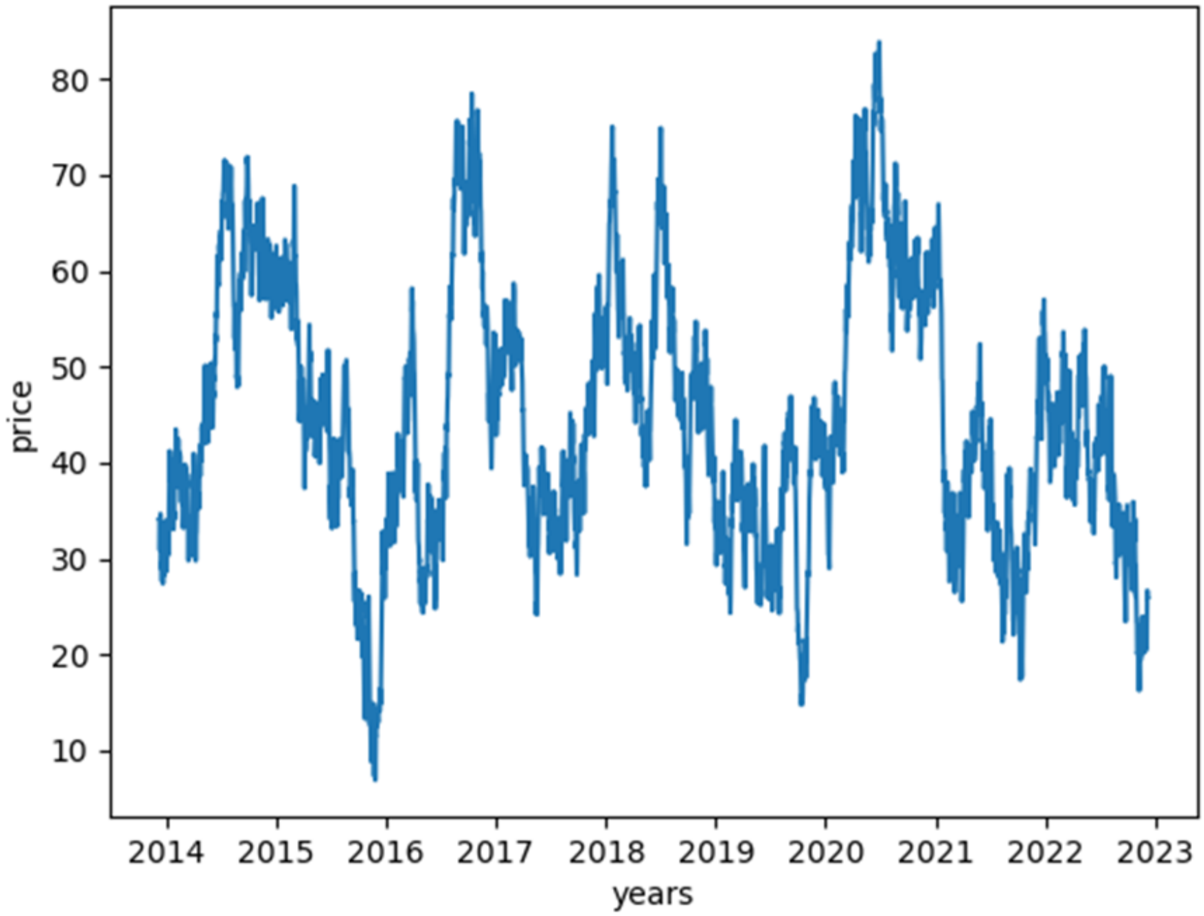

Figure 9: The generated synthetic stock of 9 years as a function of the 12 behavioral features. Using 'Function 1' which is a sum of Sin functions with different frequencies of a linear function of our behavioral features.

4.5. Synthetic data 2

The second synthetic data was based on the same 60 days history, but the function mapping behavioral features with the mean of the stock return was simpler:

$$\mu_t = \beta_1 \cdot \sin(\sum_{i=0}^{5} gm_{t,i}) + \beta_2 \cdot \sin(\sum_{j=0}^{5} pr_{t,j})$$

$$close_t = open_t + N(\mu_t, \sigma_{const})$$

Where $\beta_1 = 0.8$, $\beta_2 = 0.3$

The Resulting simulated stock can be seen in Figure 10:

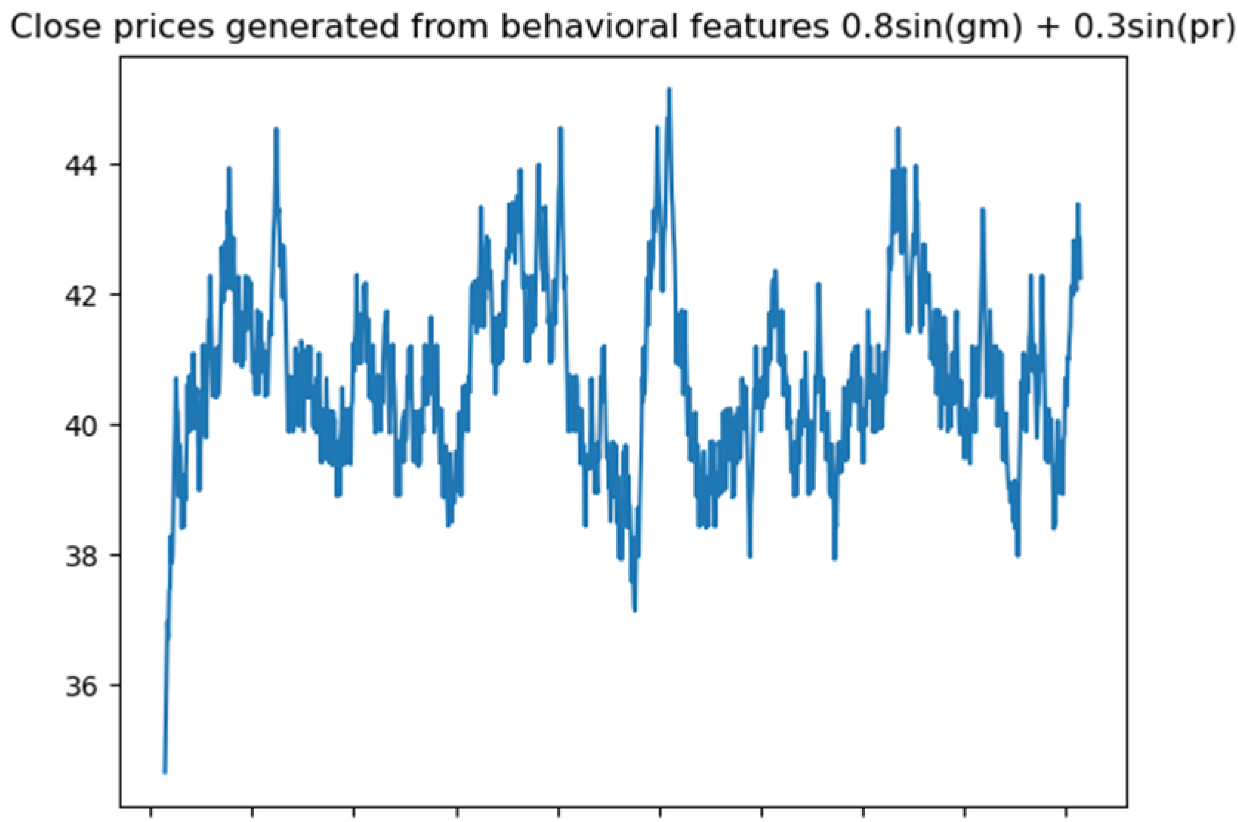

Figure 10: The generated synthetic stock of 9 years as a function of the 12 behavioral features. Using the simplified function, x-axis is years and y-axis is price.

4.6. Results on synthetic data

We trained and tested our model on both synthetic datasets, using mini-batch Adam (Diederik and Jimmy [29]) with a batch size of 1,024 and a learning rate of 0.005 for synthetic data 1 and a learning rate of 0.01 for synthetic data 2, and we tuned the rest of the hyperparameters the number of hidden units (U), lag size (T), and weight of regularization term (λ) on the validation set to optimize Adv-Psychological-ALSTM model.

4.6.1. Results on synthetic data 1

The best result was received with parameters U=20, λ=0.01, and a learning rate of 0.005. Input feature $x_t^{synthetic} \in R^{23}$ with the 11 technical features and our 12 behavioral features.

Model	Acc	MCC
Base	49.96	0.059
Psycho	50.09	0.036
Behavioral	51.49	0.081

Table 3: accuracy and MCC scores on Base model (input without behavioral features) and Behavioral (input contains an addition of 12 behavioral features)

The difference between Base and Behavioral models is the input feature $x_t^{synthetic}$. Where in the Base model it only contains 11 technical features while in Behavioral it also contains the 12 behavioral features.
Psycho model is Adv-ALSTM-Psycho model, the input is the 11 technical features with an addition of 2 behavioral features GM and PR calculated based on the mean holding period of our synthetic data.

We notice a 3% relative improvement in accuracy between Base and Behavioral models. And a negligible relative improvement in accuracy between Base and Psycho.

4.6.2. Results on synthetic data 2

The best result was received with parameters U=16, λ=0.001, and a learning rate of 0.01. Input feature $x_t^{synthetic} \in R^{23}$ with the 11 technical features and our 12 behavioral features.

Model	Acc	MCC
Base	55.83	0.1099
Psycho	61.29	0.2530
Behavioral	67.22	0.3673

Table 4: accuracy and MCC scores on Base model (input without behavioral features) and Behavioral (input contains an addition of 12 behavioral features)

Base, Psycho and Behavioral models follow Architecture 1, with the difference of the input feature $x_t^{synthetic}$. Where in the Base model it only contains 11 technical features while in Behavioral it also contains the 12 behavioral features. However in Psycho it contains the 11 technical features with an addition of 2 behavioral features GM and PR calculated based on the mean holding period of our synthetic data.

We notice a 20% relative improvement in accuracy between Base and Behavioral models. We also notice a 9% relative improvement between Base and Psycho models.

5. Discussion

In this study, we aimed to demonstrate that retail investors are sensitive to the probability of regret and grand mean which are determined by their own holding periods, and by incorporating this information into a stock movement prediction model we can improve the model's performance.
We tested different approaches in incorporating our behavioral features on real-world data and synthetic stock data.
We used four different approaches to integrate the new behavioral features to the network. While each approach yielded different results, we can see that approach "uniformly distributed", which does not consider the distribution of investors, consistently produces lower accuracy and MCC scores compared to the other approaches. On the other hand, the "weighted behavioral features" approach, which incorporates two features based on the weighted average of the behavioral features of GM and PR with respect

to the investors' distribution, performed the best in both datasets with shuffled train and validation sets. In addition, this approach has the best results for KDD17 when the sets were not shuffled. In ACL18 without shuffling, the "embeddings" approach receives the best accuracy and MCC scores. Note that embeddings is also a type of weighting for the features, only that the model learns those weights through training. These results may suggest that allowing the network to account for the different groups of investors and weigh them appropriately, is a good idea. Yet, it should be noted that the model that does no feature discrimination but assumes a "representative investor" [11] also performs relatively well, outperforming the uniformly distributed approach in terms of both accuracy and MCC scores on both datasets; this interesting behavior should be further examined in future work. Additionally, approaches that consider the distribution of investors tend to outperform both "representative investor" and "uniformly distributed" approaches, reinforcing our claim that retail investors are sensitive to our behavioral features that are determined by their own holding period.

More generally, looking at the result table, it may appear initially that the real-world data results do not demonstrate the efficacy of behavioral features. However, upon close examination of the data, the base model exhibits an accuracy of approximately 50%. Given this baseline, the observed relative improvement in performance suggests that the incorporation of behavioral features did indeed contribute to an enhancement in model performance.

Synthetic data 1 results show a 3% improvement in accuracy for Behavioral and a 20.4% improvement when the simplified function is used (synthetic data 2). Furthermore, the comparison between our approach and the "representative investor" approach, as shown in tables 3 and 4, indicates that our method yields higher accuracies in both synthetic datasets, supporting our claim that retail investors are sensitive to the probability of regret and grand mean which are determined by their own holding periods. Therefore we suggest that the findings on the ACL18 and KDD17 real-world datasets do not contradict our theory, but rather suggest potential avenues for further research. One possible explanation for these results is that the current model architecture may not be suitable for learning the complex function of behavioral features in real-world data. Even with synthetic data 1, in which we know there's a very clear relationship between stock movements and the behavioral features, the results are highly similar to the results of real-world data, indicating that the poor results in real-world data could in theory be consistent with the same strength of signal of the behavioral features.

Although the results of the current study are not necessarily inconsistent with our theory, the results of the real world data experiments were displeasing. This may be a result of several working assumptions we made due to the limitation of the available data:

1. We assume that investors start being sensitive to PR and GM features from the moment they first start holding the stock.

It is possible that investors who are passively observing the stock's performance before making the decision to purchase it for the first time may also be impacted by PR and GM, but data on such investors is not available.

2. We assume that the distribution calculated from the platform's private data is representative of investors outside of the platform.

This is a strong assumption based on the limited data we have.

3. We assume the distribution is applicable in different time periods. The distribution calculated from the platform's private data is limited to a 6-weeks period starting on January 1st 2019. While the benchmarks contain historical data from 2007 to 2016.

This is also a strong assumption based on the limited data we have.

These limitations may have affected our results on real-world data. Future work with more comprehensive data may confirm or refute this. However, it is worth noting that even with these strong assumptions, we still observe relative improvement in our real-world data.

In future research, we recommend the following approaches:

1. The use of a larger and more complex model architecture to better capture the complex relationship between investor actions and stock movements.

As explained above, the architecture could not fully capture the behavioral features signal on both real-world data and synthetic data 1 while it was able to capture it with synthetic data 2 since the relationship between stock price and behavioral features was not complex. Hence, the suggestion of a larger and more complex architecture.

2. The development of an architecture that achieves an accuracy of 60% or greater on base technical stock data without behavioral features.

With an accuracy of 60% or higher on the Base model, we can be more sure if the model's performance increased or not after adding the behavioral features since the Base performance is far from chance.

3. The acquisition of more comprehensive data on investors, with time periods aligned with stock data time periods.

The distribution of the retail investors continuously changes over time as it is the nature of the stock market, therefore having this information could contribute to our behavioral features signals in the data, evidently improving the model's performance.

4. Examination of the relative importance of incorporating behavioral features into the prediction model by analyzing scenarios that involve varying degrees of retail investor participation within the market.

Figure 1 illustrates that the percentage of retail investors in the market was 10.1% in 2010, 20.0% in 2020, and 23.0% in 2021. To effectively analyze the relative importance of incorporating behavioral features into the prediction model. We should split our dataset into multiple datasets based on the historical portion of investors in the market. Once we have each dataset, we can incorporate behavioral features and train and test our model. The benefit of incorporating behavioral features is likely to increase as the proportion of retail investors involved in trade activity grows larger.

5. Model individual investor behavior.

Modeling individual investor behavior (buying, selling, holding) in the available six weeks of data that we have from the investment platform. By doing so, we evaluate the credibility of our behavioral features in investor's behavior. This model can serve as a preliminary "stage 1" component, which can be subsequently integrated into a stock prediction model that accounts for user behavior in the market as a whole.

6. References

[1] F. Feng, H. Chen, X. He, J. Ding, M. Sun, T.S. Chua. "Enhancing Stock Movement Prediction with Adversarial Training", in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19)*, arXiv:1810.09936v2, 2019

[2] Y. Xu, S. Cohen. " Stock Movement Prediction from Tweets and Historical Prices", in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Melbourne, Australia, 2018, pp. 1970–1979

[3] D. Nelson, A. Pereira, R. Oliveira. " Stock market's price movement prediction with lstm neural networks", in *IJCNN*, Anchorage, AK, USA, 2017, pp. 1419-1426

[4] Y. Qin, D. Song, H. Chen, W. Cheng, G. Jiang, G. Cottrell. "A Dual-Stage Attention-Based Recurrent Neural Network for Time Series Prediction", in *IJCAI*, 2017, pp. 2627-2633. arXiv:1704.02971

[5] L. Zhang, C. Aggarwal, G. Qi. "Stock Price Prediction via Discovering Multi-Frequency Trading Patterns", in *ACM SIGKDD*, 2017, pp. 2141-2149.

[6] R.J. Hyndman, G. Athanasopoulos. "Forecasting: principles and practice". OTexts, 2018

[7] Z. Zhou, M. Gao, Q. Liu, H. Xiao. "forecasting stock price movements with multiple data sources: Evidence from stock market in China", *Physica A: Statistical Mechanics and its Applications*, vol 542, issue 123389, 2020

[8] W. Jiang. "Applications of deep learning in stock market prediction: recent progress", *Expert Systems with Applications*, vol 184, issue

115537, 2021.
https://github.com/jwwthu/DL4Stock

[9] Z. Jin, Y. Yang, Y. Liu, (2019, September) "Stock closing price prediction based on sentiment analysis and LSTM", *Neural Computing and Applications*, vol 32, issue 13. Available: https://doi.org/10.1007/s00521-019-04504-2

[10] Plonsly O., Erev I., Hazan T., Tennenholtz M. Random Forest: Predicting Human Behavior, in *Proceedings of the AAAI Conference on Artificial Intelligence*., 2017

[11] O. Akiva. " Predicting the behavior of investors and of stocks with machine learning and behavioral theories", MSc thesis, Dept. Economics, Technion - Israel Institute of Technology, Haifa, 2022

[12] I. Erev, E. Haruvy. "Learning and the Economics of Small Decisions", *The handbook of experimental economics*, 2, pp. 638 - 700, 2013

[13] D. Kliger, O. Levy, D. Sonsino. "On absolute and relative performance and the demand for mutual funds - experimental evidence", *Journal of Economic Behavior & Organization,* vol. 52, issue 3, pp. 341-363, November 2003.

[14] N. Barberis, R. Greenwood , L. Jin, A. Shleifer. "Extrapolation and bubbles.", *Journal of Financial Economics*, vol. 126, issue 2, pp. 203-227, August 2018.

[15] I. Erev, A. Roth. (July 2014). "Maximization, learning, and economic behavior", *PNAS*, vol. 111, supplement 3, pp. 10818-10825. Available: https://doi.org/10.1073/pnas.1402846111

[16] L. Rosasco, E. D. Vito, A. Caponnetto, M. Piana and A. Verri. (May 2004). "Are Loss Functions All the Same?," in *Neural Computation*, vol. 16, no. 5, pp. 1063-1076. Available: doi: 10.1162/089976604773135104.

[17] I.J. Goodfellow , J. Shlens, C. Szegedy. (December 2014). "Explaining and harnessing adversarial examples." arXiv preprint arXiv:1412.6572.

[18] A. Kurakin, I. Goodfellow, S. Bengio. (November 2016). "Adversarial machine learning at scale." arXiv preprint arXiv:1611.01236

[19] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhand, I. Goodfellow, R. Fergus. (December 2013). “Intriguing properties of neural networks.” arXiv preprint arXiv:1312.6199

[20] E.F. Fama, K.R. Kenneth. “Size, Value and momentum in international stock returns.”, *Journal of Financial Economics*, vol 105, issue 3, pp. 457-472, September 2012.

[21] A. Graves, A. Mohamed, G. Hinton. “Speech recognition with deep recurrent neural networks.” in *ICASSP*, 2013, pp. 6645-6649, doi: 10.1109/ICASSP.2013.6638947.

[22] L. Wu, C. Quan, C. Li, D. Ji. “Let Strangers Speak Out What You Like”, in *ACM CIKM*, New York, NY, USA, 2018, pp. 677-686

[23] J. Chen, C. Ngo, F. Feng, T. Chua. “Deep Understanding of Cooking Procedure for Cross-Modal Recipe Retrieval”, in *ACM MM*, New York, NY, USA, 2018, pp. 1020-1028.

[24] K. Cho, M. Van, C. Gulcehre. “Learning phrase representations using rnn encoder-decoder for statistical machine translation.”, in *EMNLP*, 2014, pp. 1724-1734.

[25] S. Hochreiter, J. Schmidhuber, "Long Short-Term Memory, in *Neural Computation*, vol. 9, issue. 8, pp. 1735-1780, November 1997, doi: 10.1162/neco.1997.9.8.1735

[26] C. Sengupta. “Financial modeling using Excel and VBA”. John Wiley & Sons, 2004

[27] A. Ermogenous. “Brownian motion and its application in the stock market”. 2006. Available: https://ecommons.udayton.edu/mth_epumd/15

[28] G. A. Musgrave. “A random walk down wall street”, “*Business Economics*”, vol. 32, issue. 2. April 1997

[29] D. P. Lingma, J. Ba. “Adam: A method for stochastic optimization”, in *ICLR*, San Diego, 2015. arXiv preprint arXiv:1412.6980

[30] R. Gottwald. (2012). “The use of the P/E ratio to stock valuation”, *GRANT journal*. IGA 31/2012, ISSN 1805-062X, 1805-0638 (online), ETTN 072-11-00002-09-4, pp. 21-24

[31] M. Naved, P. Srivastava, “The profitability of five popular variations of moving averages on indian market index S&P CNX Nifty 50 during January 2004 - December 2014”, *Advances in Economics and*

Business Management (AEBM), vol. 2, issue 1, pp.27-32, January-March 2015.

[32] I. Erev, E. Ert, A.E. Roth. “A choice prediction competition for market entry games: An introduction.”, *Games 2010,* vol. 1, issue 2, pp. 117-136, May 2010.

[33] I. Erev, E. Ert, A. E. Roth, E. Haruvy, S. M. Herzog, R. Hau, R. Hertwig, T. Stewart, R. West, C. Lebiere. “ A choice prediction competition: Choices from experience and from description.” *Journal of Behavioral Decision Making*, vol. 23, issue 1, pp. 15-47, December 2009.

[34] I. Erev, E. Ert, O. Plonsky, D. Cohen, O. Cohen. “From anomalies to forecasts: Toward a descriptive model of decisions under risk, under ambiguity, and from experience.”, *Psychological Review*, vol.124, issue 4, pp. 369-409, 2017

[35] O. Plosnly, R. Apel, E. Ert, M. Tennenholtz, D. Bourgin, J. C. Peterson, D. Reichman, T. L. Griffiths, S. J. Russell, E. C. Carter, J. F. Cavanagh, I. Erev. “Predicting human decisions with behavioral theories and machine learning”, 2019, arXiv preprint arXiv:1904.06866

[36] X. Zhou. Z. Pan, G. Hu, S. Tang, C. Zhao.(2018) “Stock market prediction on high-frequency data using generative adversarial nets.”*Hindawi Mathematical Problems in Engineering,* vol. 2018, article id 4907423, Available: https://doi.org/10.1155/2018/490742

[37] N.N. Taleb, The black swan, the impact of the highly improbable, United States, New York: Random House, 2007

[38] U.B. Zion, I. Erev, E. Haruvy, T.Shavit. “Adaptive behavior leads to under-diversification.” Journal of Economic Psychology, vol.31, issue 6, pp. 985-995

[39] B.F. Skinner, Science and human behavior, United States, New york: Free Press, 1953.

תקציר

ניבוי התנהגות אנושית הינו אתגר קשה עבור חוקרים שונים, בתחום כלכלה התנהגותית הצליחו להדגים את החשיבות של הטיות התנהגותיות בתהליכי קבלת החלטות אצל בני אדם. בשנים האחרונות מודלי למידת מכונה צברו פופולריות רבה בתחומים שונים בשל יכולת הלמידה הגבוה שלהם והיכולת שלהם לבצע משימות מורכבות. מחקרים אחרונים הראו את הפוטנציאל של תובנות התנהגותית לשפר את דיוק הניבוי של מודלי למידת מכונה כאשר משלבים אותם לתוך המודל. מחקר זה מרחיב על גבי מחקר קודם שמראה את היתרונות של שילוב תובנות התנהגותיות בתוך מודל פרדיקציה לתנועות מניות.

למשקיעים לא מוסדיים יש השפעה לא טריוויאלית על מחירי המניות, הם יכולים להשפיע על הסנטימנט בשוק, ובכן דרך הפעולות שלהם בשוק ומחירי המניות הם מגדירים את הטון בשוק המסחר. אפליקציות המסחר נתנו למשקיע הלא מוסדי נגישות רבה לשוק, הנגישות הזו ביחד עם דפוסי מסחר קצרי טווח והרגישות שלהם למחירי המניות עקב כך שהם משקיעים בכספם האישי, מתבטא בתגובות מהירות בשוק עקב הודעות על רווחים או כל ידיעה אחרת של חברות מונפקות בבורסה אפילו אם הידיעה לא פיננסית. ובכן ניתן להסיק שאם ננבא את התנהגות המשקיעים בשוק אזי נוכל לנבא את תנועות מחירי המניות בשוק בצורה טובה יותר, אך ניבוי התנהגות המשקיע הלא מוסדי היא משימה קשה מאוד, ואין לנו מספיק נתונים על המשקיעים ועל הפעולות שלהם בשוק, וכמו כן אנחנו לא יודעים להעריך את הפונקציה המקשרת בין פעולות משקיעים לא מוסדיים שונות לתנועות מחירי המניות בשוק. במקום זאת, אם נצליח להכניס את ההתנהגות של משקיעים לא מוסדיים לתוך מודל פרדיקציה לתנועות מחירי המניות כפיצ'רים, אז נוכל לשפר את הדיוק שלו. לפני שנמשיך להרחיב את הרעיון, קודם צריך להבין מאיפה משיגים פיצ'רים כאלה.

נעזר בספרות החלטות מניסיון, בספרות הזו הסוכן מסתמך על הניסיון שצבר עד כה מניסיונות דומים למצב הנוכחי שלו על מנת לקבל החלטה. בספרות זו הגיעו לתובנה הבאה: תמריצים כלכליים מכתיבים את הניסיון של הסוכן, הניסיון הזה מכתיב התנהגות עתידית.

לרוב מחקרים של החלטות מניסיון כרוכים בבחירה חוזרת בין שתי אופציות כך שהתפלגות הרווחים מכל אופציה לא ידועה אך הסוכנים מקבלים פידבק על החלטות שלקחו בעבר.

שוק המסחר הוא סביבה דומה לסביבה המלאכותית שמשתמשים בה במחקרים אלה: המשקיע כל הזמן לוקח החלטות במניות שלו בהתבסס על הפידבק שהוא מקבל כאשר הפידבק הוא מחירי המניות בשוק. לכן הסתמכות על תוצאות מחקרים אלה היא תוצאה טבעית.

מחקרים אלה טוענים, מודל ניבוי טוב עבור משימות חוזרות צריך להתחשב בשני דפוסי התנהגות שאנשים מראים רגישות רבה אליהם:

1. התמורה הממוצעת שהושגה עד עכשיו מכל אופציה
2. ממוצע מדגם קטן של האופציות שהיו מניסיונות מהעבר.

בהתבסס על תיאוריות התנהגותיות בתחום החלטות מניסיון, הנחנו שדפוסי המסחר של המשקיע הלא מוסדי מושפעים מהרגישות שלהם לתשואה ממוצעת של המניה ומסיכוי החרטה הנובע מהחזקת המניה. בנוסף לכך, הבחנו שמשקיעים לא מוסדיים מחולקים לקבוצות שונות בהסתמך על תקופת ההשקעה שלהם, חלק מהמשקיעים הם משקיעי יום (כלומר מחזיקים במניה לתקופה של יום ולא יותר), חלק שבועי וחלק חודשי. לכן בנינו את ההשערה שאומרת שבגלל שקבוצות שונות של משקיעים חווים דרגות שונות של חרטה וממוצעי תשואה שונה, אז אם נספק את המידע הזה למודל ניבוי תנועת מניות ע"י הוספת פיצ'רים (נקרא להם פיצ'רים התנהגותיים) המתבססים על הערכות לסיכוי החרטה ותשואה ממוצעת לקבוצות השונות של המשקיעים אז נוכל לשפר את דיוק ניבוי המודל. בדקנו את ההשערה שלנו בעזרת דאטא אמיתי של מחירי מניות וגם דאטא סינטטי. התוצאות מעידות שכאשר הסיגנל של הפיצ'רים ההתנהגותיים חזק מספיק אז ההתחשבות בהם עוזרת למודל. אך, התוצאות מעידות גם שדאטא אמיתי של מחירי מניות מכיל יותר מדי רעש מכדי שהפיצ'רים יוכלו לתרום מהותית.

המחקר הזה נעשה בפקולטת מדעי הנתונים וההחלטות בהנחיית ד"ר אורי פלונסקי ופרופסור מרגריטה אוסדצ'י.

מחברת חיבור זה מצהירה כי המחקר, כולל איסוף הנתונים, עיבודם והצגתם, התייחסות והשוואה למחקרים קודמים וכו', נעשה כולו בצורה ישרה, כמצופה ממחקר מדעי המבוצע לפי אמות המידה האתיות של העולם האקדמי. כמו כן, הדיווח על המחקר ותוצאותיו בחיבור זה נעשה בצורה ישרה ומלאה, לפי אותן אמות מידה.

אני מודה לטכניון על התמיכה הכספית הנדיבה בהשתלמותי.

ניבוי של תנועת מניות באמצעות פיצ'רים התנהגותיים

חיבור על מחקר

לשם מילוי חלקי של הדרישות לקבלת תואר מגיסטר למדעים
במדעי נתונים ומידע

ראניא ח'ורי

הוגש לסנט הטכניון - מכון טכנולוגי לישראל

טבת התשפ"ג, חיפה, דצמבר 2022