An Enriched Time-Series Forecasting Framework for Long-Short Portfolio Strategy

XIURUI HOU1, KAI WANG2, JIE ZHANG3, (Student Member, IEEE), AND ZHI WEI1, (Senior Member, IEEE)

1Department of Computer Science, New Jersey Institute of Technology, Newark, NJ 07102, USA
2School of Business, Stevens Institute of Technology, Hoboken, NJ 07030, USA
3Adobe Inc., San Jose, CA 95110, USA
Corresponding author: Zhi Wei (zhiwei@njit.edu)

ABSTRACT Stock return forecasting typically requires a large number of factors and these factors usually exhibit nonlinear relations with each other. Conventional methods of stock return forecasting mainly fall into two categories: Technical Analysis and Fundamental Analysis. Technical Analysis focuses on time-series data, while Fundamental Analysis explores low-frequency fundamental variables. Although there are substantial works on either time-series analysis or fundamental analysis, few studies have enriched the time-series forecasting with fundamental variables, as the features are characterized by different frequencies, scales and types. In this paper, we propose a Long Short-Term Memory and Deep Neural Network (LSTM-DNN) hybrid model to integrate the fundamental information into time-series forecasting tasks. We demonstrate how investors can benefit from the superior performance of LSTM-DNN by constructing a long-short portfolio that takes long positions in stocks with the highest forecasting returns and short positions in stocks that are expected to decline. Extensive experimental results on real data show that our novel framework could improve the profitability of long-short portfolio strategies compared to the state-of-the-art approaches. We also find evidence indicating that the outperformance of LSTM-DNN model comes from its enhanced ability to extract information from the nonlinear relations among various features, rather than bearing more market risks. Besides the novel framework, we propose a cross-section normalization method, which benefits the framework by providing enriched cross-section signals.

INDEX TERMS Hybrid deep neural networks, long-short portfolio strategy, security returns forecasting.

I. INTRODUCTION Accurate stock return forecasting and clear elucidation of the underlying causal factors are of interest to various parties. However, it is still a notoriously challenging issue owing to the complex, dynamic, and chaotic nature of the stock market. Despite the challenges, many empirical studies have shown that financial markets are predictable to some extent [1]–[3].

Within the past decades, we have witnessed a dramatic increase in researches on stock return prediction. Technical Analysis (TA) and Fundamental Analysis (FA) are two popular methods in the literature of stock return forecasting. TA focuses on modeling time-series price data and predicts future stock prices primarily based on historical price trends. The development of the TA method is mainly driven by data mining and machine learning fields. Various statistical and machine learning methods have been explored and applied, such as Auto-Regression, Neural Networks, Support Vector Machine, and Random Forest. However, the complex and non-stationary property of stock returns imposes a formidable challenge for most time-series modeling approaches. In contrast, FA is a classical topic in finance, especially in asset pricing, and is generally considered for developing long-term investment strategies. FA pays more attention to fundamental factors and variables that are extracted from the firms’ annual reports and other announcements. Both TA and FA have their unique advantages on stock return forecasting.

Recently, several attempts were carried out to integrate TA and FA. They propose to first aggregate the time-series sequence into TA indicators (e.g., Moving Averages, Trend Lines, and Relative Strength Index) and then train the predictive models based on both TA indicators and FA variables. It is not difficult to prove that the temporal dependency extracted shrinks into the ad-hoc TA indicators. Integrating...
both types of analyses while preserving the superiority of the time-series model remains a challenging task due to three main reasons: Firstly, the types of data to deal with are quite different. TA usually deals with time-series data, while FA focuses on annual or quarterly fundamental data. Secondly, TA and FA use different analyzing tools. Classical TA models use moving average and auto-regression, while sorting stocks based on firm characteristics into portfolios is the conventional method in FA. Finally, they differ in their trading strategies. TA attempts to find better trading signals on one or two assets and simply applies a pair trading or buy-and-hold strategy. In contrast, trading strategies developed by FA require a long-short portfolio, which longs the stocks with higher expected returns and shorts the stocks with lower expected returns. This paper proposes a novel framework by blending the Long Short-Term Memory (LSTM) and Deep Neural Network (DNN) to capture the temporal dependencies among variables and integrate stable fundamental information, respectively. We call our blended framework the LSTM-DNN model thereafter in this paper.

In recent years, advanced machine learning methods received much more attention due to their advantage in capturing the nonlinear relations among various factors compared with linear models. Traditional econometric methods, such as ordinary least square (OLS) and panel regressions, are popular in economic and finance studies, as their results can be easily interpreted and validated. To the opposite, advanced machine learning techniques, e.g., Deep Neural Networks (DNNs), are capable of solving complex tasks, e.g., image recognition, natural language processing, but at the expense of interpretability. To enjoy more precise predictions and address potential limitations, we study the improvement of LSTM-DNN by feeding it a different set of features and comparing it to the benchmarks.

To incorporate both time-series data and fundamental information, our blended framework models them separately and then combines the intermediate results to make final predictions. Specifically, LSTM takes care of the time-series data and aims to bring the time-series flavor to the last layer of neurons of the model; Fundamental data are handled by DNN which allows for more complex relations to exist among various features. From the perspective of the profitability of the strategy, DNN and LSTM are able to extract richer information from features relative to ordinary econometric models which are developed based on a bunch of assumptions. About the trading strategy, we limit ourselves in a long-short portfolio strategy in this paper, while other strategies are applicable.

Intuitively, the long-short portfolio strategy can achieve its best performance when we accurately locate a stock’s expected return on the entire cross-section of the expected returns, not necessarily its exact expected return. Thus, rather than using raw data, we apply a cross-section-aware preprocessing on features and feed them to our LSTM-DNN framework. Specifically, we standardize all individual stock features within each month by subtracting the cross-section average value of a feature, except for categorical features, and dividing the resulting value by the cross-section standard deviation of the given feature. For example, if the lag 1-month return of a stock is 1% while the average stock return in that month is 2%, and the cross-section standard deviation is 2%, the lag 1-month return of the given stock after cross-section preprocessing is $(1% - 2%)/2% = -0.5$. Via such a preprocessing, the factors encompass richer information in the sense that it contains the relative position information of the firm characteristics across all firms within each month. It is also important to note that all factors we use are available at the time of portfolio construction.

The long-short portfolio strategy works as follows. We rank stocks based on their individual predicted returns. Then, given their ranks, we buy the top decile\(^\text{1}\) and sell short the bottom decile of securities once every rebalancing period. One advantage of the long-short portfolio strategy is that the movement of the whole market has little impact on the performance of this strategy. For example, a hedge fund might sell short $1 million of Apple stock and buy $1 million of Microsoft stock. With this position, any event that causes all tech industry stocks to fall will make a profit from the Apple position and bear a matching loss on the Microsoft position. By exploiting a long-short portfolio, namely, invest the stocks with the highest predicted returns and sell short the stocks with the lowest predicted returns, an investor can profit from inferring the relative performance of a stock in the future period.

To conduct a robustness check on the long-short portfolio based on LSTM-DNN model, we still focus on the rate of return and/or Sharpe Ratio (rate of return divided by its standard deviation) delivered by the strategy which is consistent with most trading strategy developing papers [4], [5]. It is noted that the conventional metrics commonly used in data mining, such as mean square error (MSE), may not be a good indicator for performance evaluation.

The layout of the paper is as follows. Section 2 reviews the related literature of machine learning approaches in individual stock return prediction. Section 3 describes the problem formulation and predictive modeling. Section 4 introduces the data and experiment setting. Section 5 examines the performance of the proposed method and other machine learning methods. The robustness check and feature importance discussion are delivered in section 6. Section 7 concludes and discusses areas for future works.

II. RELATED WORK

A. ADVANCEMENTS MADE IN FINANCE FIELDS

Trading strategies have been extensively explored in finance literature. Many of them are proved to be effective in forecasting stock returns [6]. Value investing has become popular in professional financial advisers and practitioners since the famous book “The Intelligent Investor” [7] was published in 1965. After that, many fundamental factors have been studied. These factors span a wide range of information,\(^\text{1}\) Each part represents 1/10 of the sample or population.
such as market capitalization [8], price-to-earnings ratios [9], trading liquidity [10] and return momentum [11]. Many of these factors exhibit unique predictive value for forecasting a stock’s future performance. In particular, momentum has received a tremendous amount of attention from practitioners [12]–[14]. Many follow-up studies have been focused on optimizing the momentum trading strategy. For example, a two-pass ordinary least squares regression method has been proposed to extract more information from a stock’s historical performance [15].

**B. CLASSICAL MACHINE LEARNING METHODS**

Forecasting is a classical machine learning problem. Quite a few conventional machine learning methods have been developed in data mining and machine learning fields for predicting stock returns. For example, Support Vector Machine (SVM) was used to forecast the weekly movement direction of NIKKEI 255 index [16] and predict the stock market trend of Taiwan stock market [17]. Random forest was employed to predict the direction of stock market prices and was found to outperform SVM and Artificial Neural Networks (ANNs) in empirical experiments [18]. More recently, a combined SVM and K-nearest neighbor with weighted features have been proposed to predict Chinese stock market indices [19].

**C. DEEP LEARNING APPROACHES**

Neural networks, which achieved the best performance in multiple areas [20], have been applied to analyze stock markets. Shallow artificial neural networks (ANNs) were used to predict stock price movement for firms traded in China [21] and Canada [22]. These ANNs have shown more predictive power than linear models. The past few years have witnessed a surging number of the applications of deep neural networks (DNNs) for forecasting financial markets. Abe and Nakayama [23] used DNNs to predict one-month-ahead cross-section stock return. Various DNN structures, with differing numbers of layers and differing numbers of neurons in each layer, were investigated. They found, the more layers, the higher the directional accuracy would be. DNNs, gradient-boosted trees, random forest, and different ensembles have been evaluated for predicting the one-day-ahead stock return on the S&P 500 [24]. The predicting target is a binary response indicating if the stock return is larger than the corresponding cross-section median return computed over all stocks. Their results showed that a simple equal-weighted ensemble of neural networks, gradient-boosted trees and random forests achieved the best performance. A recent study also compared different structures of DNNs and other advanced machine learning approaches. But it considered more extensive features, including 94 characteristics of each stock, the interactions of these characteristics, and eight aggregate time-series variables [5]. They found shallow neural networks outperformed other competing methods. There are also some studies concerning semantic or sentiment information in optimization of asset allocation and volatility forecasting [25], [26].

Long Short-Term Memory Networks (LSTM) have been used to predict future trends of stock prices [27] and directional movements [28]. Both studies conclude that the LSTM, a better alternative for time-series problems, outperforms plain deep neural networks and classical machine learning methods (e.g., random forest).

Most of these existing methods are TA-based, using pure historical price/return [24], [27], [29]–[31] or adding derived technical indicators [17], [18], [32], [33]. FA-based approaches are less common (e.g., [22] used only fundamental information). A few works have tried to integrate TA and FA [5], [21]. However, they ignore temporal dependency and simply mix time-series variables with other variables. Such methods fail to exploit temporal dependency information and are usually sub-optimal.

**D. HYBRID MODELS**

Besides single-mode machine learning algorithms, there are some hybrid models proposed in recent years. For example, the wavelet technique is integrated with neural networks to forecast stock markets [29], [32]. A hybrid model of ANNs and autoregressive moving average models is proposed [30]. An ensemble of technical analysis and machine learning method is proposed in [34]. More hybrid models for stock market forecasting can be found in [31], [33], [35]–[37]. Recently, the hybridization of different kinds of neural networks, which can accommodate heterogeneous input features, has been applied to other areas successfully, e.g., success prediction on crowdfunding [38] and user intended action prediction [39]. These promising results motivate us to blend different kinds of neural networks for characterizing the heterogeneous input features to predict stock returns.

**III. METHOD**

**A. PROBLEM FORMULATION**

We aim to predict the monthly return for each stock based on the historical data. Let subscript $t$ and $t - \tau$ denote the $t$-th month and lagged $\tau$ month, respectively. In addition, we use $r$ and $\hat{r}$ to represent actual stock return and the predicted stock return throughout the paper. We denote the operable stock set at time $t$ as $\mathbf{\Omega}_t$. Note that the stocks with missing values in the required features are not included in the set $\mathbf{\Omega}_t$. Let $r_i$ be the monthly return of the $i$-th stock at time $t$, where $i \in \{1, \ldots, N_t\}$ and $N_t = |\mathbf{\Omega}_t|$ represents the number of active stocks at month $t$. Note that the number of active stocks, $N_t$, might vary at each time point due to the missing data. For the stock fundamental information, $f_{ij}$ represents the latest available annual fiscal report statistics of stock $i$ at time $t$. Assume that there are $M$ fundamental features in total. We denote by $f_{ij}$, where $j \in \{1, \ldots, M\}$, the $j$-th fundamental feature of stock $i$ at time $t$.

**B. CROSS-SECTION STANDARDIZATION**

We propose to construct the long-short portfolio based on the predicted relative performance of each stock. Instead
TABLE 1. Feature notations and descriptions.

| Feature | Description |
|---------|-------------|
| \(r^t_{−1}, \ldots, r^t_{−18}\) | Lagged 1 to lagged 18 monthly return |
| PRC\(_{−1}\) | Lagged 1-month price |
| VOL\(_{−1}\) | Lagged 1-month volume |
| SIZE\(_{−1}\) | Lagged 1-month size (price*shares outstanding) |
| BM\(_h\) | The book-to-market ratio in the last fiscal year |
| BM\(_{pos}\) | Indicator of positive book value in the last fiscal year |
| SICCC\(_D\) | The industry category |
| past6ave | Average return of the market in past 6 months |
| past24ave | Average return of the market in past 24 months |
| past6ave | Average return of the market in past 60 months |

The first feature is for LSTM and the rest are for DNN. Superscript \(i\) represents stock \(i\).

of using the raw historical stock returns and fundamental information, we normalize the dependent and independent variables across all the active stocks at each month to encode the relative information rather than an absolute term. Let \(v^t_i\) be a numerical variable of the \(i\)-th stock at time \(t\). We normalize the raw variable \(v^t_i\) and make it comparable along the time by removing the impacts of group mean and standard deviation. Specifically, the normalized value \(\tilde{v}^t_i\) is calculated as follows:

\[
\tilde{v}^t_i = \frac{\sum_{j=1}^{N^t_i} v^t_j / N^t_i}{\sqrt{\sum_{j=1}^{N^t_i} (v^t_j - \bar{v}^t)^2 / (N^t_i - 1)}}
\]

Note that the normalization is also consistent with the essence of cross-section stock return prediction, which is to accurately forecast the relative performance of a stock [40].

As shown in Table 1, for each stock, we calculate 27 features, including 18 monthly return variables and 9 fundamental variables. Excepting the average market returns and industry category (a classification code to group companies with similar products or services), which are categorical variables, all other variables are cross-section standardized before feeding to various machine learning models. In this paper, we focus on a compact number of features. A larger set of features may improve the overall performance, but at the expense of increased cost to collect the required data, especially for inexperienced individual investors. Further works will be exploring more types of features and the impacts of different features.

C. LSTM-DNN MODEL

We apply a stacked Long Short-Term Memory Network (LSTM) to model the monthly return time series \(\{r^t_1, \ldots, r^t_{18}\}\). A compact form of the equations for the forward pass of an LSTM unit with a forget gate is represented as follows:

\[
f_t = \text{sigmoid}(W_f r^t_i + W_f b_{h^t_i} - b_f),
\]

\[
b_t = \text{sigmoid}(W_i r^t_i + W_i b_{h^t_i} - b_i),
\]

\[
o_t = \text{sigmoid}(W_o r^t_i + W_o b_{h^t_i} - b_o),
\]

\[
c_t = f_t \circ c_{t-1} + i_t \circ \text{tanh}(W_c r^t_i + W_c b_{h^t_i} - b_c),
\]

\[
h_t = o_t \circ \text{tanh}(c_t),
\]

where \(\circ\) denotes the element-wise Hadamard product; \(\tilde{r}^t_i \in \mathbb{R}^t\) is the normalized input vector at timestep \(t\); \(f_t\) is the forget gate activation vector; \(i_t\) and \(o_t\) are input and output gates’ activation vectors, respectively, while \(h_t\) and \(c_t\) represent the hidden state (output vector) and cell state. \(W_{f,r} \in \mathbb{R}^{h \times t}, W_{f,b} \in \mathbb{R}^{h \times b}\) and \(b \in \mathbb{R}^h\) are weight matrices and bias vectors. In contrast, the fundamental variables are fed into a fully-connected neural network (FCNN). We concatenate the outputs from both LSTM and FCNN and add additional fully-connected layers to output the final predictions. Mathematically, the output of the hidden neuron in the fully-connected layer is defined as

\[
\sigma (\sum_{j=1}^{n^t_i} w_j x_j + b_j),
\]

where \(\sigma\), \(w_j\) and \(b_j\) represent the activation function, weight matrix and bias, respectively. Following Nair and Hinton [41], we use rectified linear unit (ReLU) activation function for all hidden neurons. The ReLU activation is defined as follows:

\[
\sigma_{\text{ReLU}} (\sum_{j=1}^{n^t_i} w_j x_j + b_j) = \max(0, \sum_{j=1}^{n^t_i} w_j x_j + b_j).
\]

The loss function we use is the mean squared error (MSE). For illustration, the mean squared error at time \(t\) is defined as follows:

\[
MSE_t = \sum_{i=1}^{N^t_i} (\tilde{r}^t_i - \hat{r}^t_i)^2 / N^t_i,
\]

where \(\tilde{r}^t_i\) is the real standardized return of stock \(i\) at time \(t\) and \(\hat{r}^t_i\) is its estimation. Figure 1 shows the structure of the proposed LSTM-DNN model together with the number of hidden nodes at each layer. Techniques such as L2 regularization and Dropout [42] are applied to avoid overfitting. Linear activation function is used for the final output layer, while the ReLU activation functions are utilized for other layers. Adam algorithm [43] with learning rate of 0.001, and decay rate of 0.001 is adopted to optimize the model.

D. LONG-SHORT PORTFOLIO

The predictive return of stock \(i\) for month \(t\) is \(\hat{r}^t_i = f((r^t_{i−1}, \ldots, r^t_{i−1}, f_{i−1}, \ldots, f_{i−1}))\), where \(i \in \{1, \ldots, N_t\}\). We denote \(f(\cdot)\) the trained LSTM-DNN model. Based on the predicted returns of each stock for the month \(t\), we rank the stocks in ascending order and re-index them as \(\{1, \ldots, N_t\}\).

The long-short portfolio is constructed as follows:

\[
\hat{\Omega}_t^{\text{short}} = \{1, \ldots, [\alpha \cdot N_t]\}
\]

\[
\hat{\Omega}_t^{\text{long}} = \{[(1 - \alpha) \cdot N_t], \ldots, N_t\},
\]

where \(\alpha \in (0, 0.5]\) is the cutting percentage and symbol \([x]\) takes the minimum integer larger than \(x\). To be consistent with existing methods in the literature, we use \(\alpha = 0.1\) in experiments and we also show that the LSTM-DNN model is superior to other methods with any \(\alpha\) in Section VI-C. \(\hat{\Omega}_t^{\text{short}}\) and \(\hat{\Omega}_t^{\text{long}}\) are the stock sets for longing (which are predicted to have higher returns) and shorting (which are predicted to have lower returns), respectively.
There are at least two strategies for investors to profit from our model. First, a simple buy-and-hold strategy that only longs the predicted ‘winner’ stocks provides investors with expected monthly returns above the expected market returns. Second, by constructing a long-short portfolio, which long the predicted ‘winner’ stocks and short predicted ‘loser’ stocks, investors can exploit the return difference between the two sides. In this paper, we focus on the second strategy, which is more common in literature.

IV. EXPERIMENT

A. DATA DESCRIPTION

The experimental data is obtained from the Center for Research in Security Prices (CRSP). Specifically, the data we downloaded from CRSP includes individual stock returns and prices, S&P 500 index return, industry categories, number of shares outstanding, share code, exchange code, and trading volume. The prices of stocks are recorded at the end of each month and adjusted for stock splits and stock dividends. The stock return of month $t$ is calculated by using the adjusted close price of month $t$ divided by the adjusted close price of month $t-1$ and then minus 1. S&P 500 return data is used to derive market index $\text{past}\{6, 24, 60\}\text{ave}$ in Table 1. Shares outstanding is the number of publicly held shares recorded in thousand. We only study the returns of common shares (with CRSP share code 10 or 11). Daily trading volumes are aggregated to calculate the monthly trading volumes. Standard Industrial Classification Code (SICCD) is used to group companies with similar products or services. In this paper, we only study stocks listed on the three normal exchanges, NYSE, AMEX, and Nasdaq (with exchange code 1, 2, and 3, respectively).

The firm’s book value data is obtained from COMPUSTAT and the one-month Treasury bill rate is obtained from Kenneth French’s data library. Book Value Per Share (BKVLP$S$) in COMPUSTAT is the Common Equity Liquidation Value (CEQL) divided by Common Shares Outstanding (CSHO) in a fiscal year-end.

B. EXPERIMENTAL SETTING

The data, ranging from July 1, 1977 to December 31, 2018, are collected from different sources. The total number of stocks in our dataset is 6,700, with approximately 1,398 operable stocks per month. It should be noted that the market structures and trading rules are evolving as investors apply mutable trading strategies, and regulators may change trading rules of the market. For example, the Dodd-Frank Act strengthened the regulation of interbank lending system after the 2008 financial crisis, and the popularization of high-frequency trading enhances the market liquidity during normal periods. These structural changes require us to treat the strategies in each period differently. We split the 30-year period from 1989 to 2018 into six 5-year testing periods. Each testing period is trained with the data of its most recent 10-year period. Within each 10-year period, the first 8-year data is used to train the model, and the last 2-year data set is selected for validation (for neural network related methods only). Thus, every two consecutive training periods are overlapped with a 5-year period. Figure 2 illustrates the data splitting procedure. The first training sample, Jan 1979, as shown in the figure, is using the historical data from Jul 1977 (lagged 18) to Dec 1978 (lagged 1) as the input feature.

We consider different scenarios by varying input features and data preprocessing procedures. As shown in Table 2, ‘S’ indicates that the inputs go through cross-section standardization preprocessing and ‘US’ indicates that the inputs
are not standardized. ‘Lag18’ and ‘FDMT’ represent the lagged 18 returns and fundamental variables, respectively. ‘*’, can be replaced by the name of each method. For example, DNN_S_Lag18_FDMT is the method in which the inputs include standardized lag 1 to lag 18 returns and latest fundamental variables and the model is Deep Neural Network.

C. BASELINE METHOD

We compare our method with following six competing baselines:

- Naive Momentum Strategy (Naive_MM) [12]
- Ordinary Least Square (OLS)
- Support Vector Machine (SVM)
- Random Forest Regression (RF).
- Long Short-Term Memory (LSTM)
- Fully-Connect Deep Neural Networks (DNN)

The conventional momentum strategy (Naive_MM), proposed by Chan et al., ranks stocks by the average of past t-2 to t-12 monthly return, and then longs the stocks in the winner decile (top 10%) and shorts the stocks in the loser decile (bottom 10%) [12]. Fitting an OLS model is another popular technique in finance analysis. During the last decade, SVM and RF have exhibited the state-of-the-art performance in both classification and regression tasks [44]–[48]. In addition, the Sharpe Ratio (SR), which is popular in finance literature, is also considered as the second evaluation metric. Formally,

\[ SR = \frac{R_p - R_f}{\sigma_p}, \]

where \( R_p \) is the average portfolio return, \( R_f \) is the 1-month treasury bill return, and \( \sigma_p \) represents the portfolio return standard deviation.

D. EVALUATION METRIC

Following previous studies, the average expected monthly return (MAR) is used to evaluate the performance of competing models [5], [19], [22], [27], [32]. Mathematically,

\[ MAR_t = \frac{\left\{ r_t^i \right\}_{i \in \hat{q}_{t}^{long}} - \left\{ r_t^i \right\}_{i \in \hat{q}_{t}^{short}}}{\sigma_p}. \]  

In addition, the Sharpe Ratio (SR), which is popular in finance literature, is also considered as the second evaluation metric. Formally, 

\[ SR = \frac{R_p - R_f}{\sigma_p}, \]

where \( R_p \) is the average portfolio return, \( R_f \) is the 1-month treasury bill return, and \( \sigma_p \) represents the portfolio return standard deviation.

V. RESULT

Following Gu et al. [5], we calculate one-month-ahead out-of-sample stock return predictions for each method at the end of each month. For each model, we sort the stocks in ascending order based on the predictions. The ordered stocks are equally divided into 10 buckets, denoted as Decile 1 - 10 (See the first column of Table 3). Specifically, Decile 1 and 10 represent the stock sets with the lowest and highest investment returns, respectively. The equity long-short strategy takes long positions in stocks that are expected to increase in value (Decile 10) and short positions in stocks that are expected to decrease in value (Decile 1) to earn the spread. The profit of the equity long-short strategy is presented in Table 3. To clearly illustrate the results, we calculate the average of actual monthly returns for the stocks in each bucket. Here we make a few remarks. Firstly, the Decile 1 and Decile 10 obtained based on the predictions from LSTM-DNN have the lowest and highest actual returns, respectively, which demonstrates that, compared with other methods, LSTM-DNN delivers more accurate monthly return predictions. Secondly, LSTM-DNN also provides better results for the middle Deciles (Decile 2-9). As shown in Table 3, Decile 2-9 obtained via LSTM-DNN have incremental returns in most cases. With standardized time series data and fundamental variables (Table 3: Standardized:Lag 18 + FDMT), LSTM-DNN could generally achieve incremental results from Decile 1 to Decile 10. In contrast, the results obtained via other methods are much worse than some predicted lower Deciles have much larger actual returns than the predicted higher Deciles (For example, Decile 8 and Decile 9 achieved by SVM have much lower actual returns than the Decile 1-2.). Thirdly, the profit of the
equity long-short strategy could be maximized by integrating the proposed LSTM-DNN method. With the standardized inputs (Section VI-B) and the proposed LSTM-DNN model, we could achieve the highest portfolio-level profit of 3.556%. Table 3 also empirically demonstrates the contribution of fundamental variables. When integrated with fundamental features, we have higher returns in Decile 10 and lower returns in Decile 1. The observation is consistent across different methods including OLS, DNN, and LSTM-DNN, strongly indicating that fundamental information could help the model make more accurate predictions and build better portfolios. Interestingly, the predicted ‘winner’ portfolio (Decile 10) constructed by LSTM-DNN framework achieves 3.329% monthly return, whereas the predicted ‘loser’ portfolio has a negative monthly return (i.e., \(-0.227\%). These results suggest that the statistical inference of LSTM-DNN contributes to the final profit from both ‘long’ and ‘short’ sides, when we apply the equity long-short strategy.

Since we use the batch sliding window to do the evaluation, Table 4 presents the monthly average return (MAR) and Sharpe Ratio (SR) for each method at different testing periods. LSTM-DNN model with cross-section standardized inputs demonstrates the best performance, measured by MAR and SR, in most batches and yields strongly comparable performance in other batches. By comparing the upper panel and lower panel, we observe that cross-section normalization procedure generally benefits all the models.

Figure 3 visualizes the cumulative profit for each method over the entire out-of-sample test period. For each method, we start trading with 1 dollar from Jan 1st, 1989 and calculate the cumulative value of the portfolio for each month. To avoid a crowded figure, all methods presented in the figure use the standardized inputs. The period of the global financial crisis in 2008 is highlighted in gray in Figure 3. As shown in Figure 3, LSTM-DNN model could achieve...
the highest cumulative return from the whole testing period (1989 - 2019). In addition, adding fundamental features makes better predictions, as all the methods with ‘Lag18_FDMT’ are more profitable than the methods with ‘Lag_18’. It should be noted that the investment experiment is conducted under simplified scenarios and external factors (e.g., transaction fee) are not considered. In the future, we’d like to build more sophisticated simulation frameworks to provide more accurate analytical results.

VI. DISCUSSION

In this section, we conduct several discussions related to the performance of the LSTM-DNN model. We first check the robustness of the performance by looking at its profitability during the worst recession periods. Then, we examine whether the cross-section normalization procedure really enhances the model performance. We also show that the arbitrary 10% cutting point does not give our LSTM-DNN model any advantage. Finally, we use a Saliency map [49], [50] to illustrate the feature importance and compare it with the results from linear and nonlinear model.

A. RISK ANALYSIS

Although our LSTM-DNN model has larger investment returns compared to the competing models, whether it is a good trading strategy for most investors is not completely determined. To clearly answer this question, we have two more aspects to examine for our strategy. First, investors have different expectations about their portfolio performance in different periods. For example, it is not acceptable for most investors if a strategy delivers extremely bad performance in the financial recession, even though it outperforms other strategies in other periods. Second, we check whether LSTM-DNN strategy has higher systematic risks. A strategy can earn much higher returns by simply leveraging its exposure to systematic risks according to efficient market hypothesis.\footnote{https://en.wikipedia.org/wiki/Efficient-market_hypothesis}

For the first point, Table 5 shows the performance of different methods during the worst 15 months measured by the S&P 500 index in our entire testing period. All three methods perform much better than the market in these months while LSTM-DNN (LD) performs the best in most months.
Our LSTM-DNN strategy delivers an average return of 3.56% per month, but the S&P500 index loses lots of money during the worst periods of the market. The worst return of the LSTM-DNN strategy during these months is \(-2.7\%\). In contrast, DNN and OLS are much worse and their returns are \(-13.63\%\) and \(-9.57\%\), respectively, for the worst months. Therefore, the LSTM-DNN model-based strategy should be more attractive to investors.

For the second aspect, we find that the risk-adjusted LSTM-DNN strategy has similar performance compared with its original returns across the whole testing period. Specifically, we regress the return of the LSTM-DNN model on the Fama-French three factors, and the resulting intercept of the regression is the risk-adjusted return of the strategy. Thus, it is very likely that the LSTM-DNN model obtains better performance by extracting more complex interactions among factors rather than simply leveraging the risk exposures to common risk factors.

### C. CHOICE OF CUTTING PERCENTAGE

To avoid any advantages that ad hoc setting can bring to the LSTM-DNN model, we calculate both the average monthly return and the Sharpe Ratio for the best performing strategies within a wide range of cutoffs. Sharpe Ratio is a more comprehensive metric, as it considers both return and volatility. From Figure 5, we observe that LSTM-DNN constantly performs better than the other models. Note that all three models reveal a similar pattern regarding both return and Sharpe Ratio. All three methods achieve better average return as the cutting percentage gets smaller, and earn an astonishing 8% monthly average return when the cutting percentage is 1%. 1% cutting percentage means we short the bottom 1% and long the top 1% of stocks. It could only achieve much higher returns if the prediction is accurate. The drawback is that the variance of a smaller set of stocks is usually larger. In contrast, the Sharpe Ratio, which further considers the volatility of the portfolio return, achieves the highest value around 10% cutting percentage for all three methods and decreases dramatically as the cutting percentage goes below 32000.
D. FEATURE IMPORTANCE ANALYSIS

Although we do not have a direct way to demonstrate the non-linear interactions among various time-series features and firm fundamental variables, we are able to measure the agreement and disagreement on the relative feature importance between the LSTM-DNN model and benchmark OLS model. Figure 6 shows the relative feature scores from the two methods across different time periods using the Saliency Map technique [51]. Overall, we find that among many factors, lag 1-month return, book-to-market ratio, and a couple of industry categories outweigh other factors in most of the periods. It is clear that even for a single method, the relative feature importance is not the same across different periods. For example, a firm within the agriculture industry is expected to earn higher returns than average level except for the most recent half-decade when it is measured by OLS model. These differences may reflect the structural change of the market caused by policymakers or investors.

The disagreement between LSTM-DNN and OLS, however, is more of our interest. For example, the LSTM-DNN model emphasizes the lagged returns compared with the OLS model for most of the periods. Intuitively, this is because the LSTM model has advantages in capturing meaningful information from time-series data. The two models also reveal disagreement on the feature importance among categorical variables. We also find a diminishing trend of factors’ importance in recent decades, which echoes to the hypothesis that the market is becoming more efficient. It is important to note that we employ the same set of information and the same data preparation procedure for all models, which guarantees the consistency of feature importance comparison.

VII. CONCLUSION

In this paper, we propose a hybrid neural network framework to integrate both time-series data and stable fundamental features for the stock return prediction. A cross-section data normalization method is introduced for enriching the signal of the relative performance of each stock. We conduct various experiments on the CRSP data and show that the proposed method could achieve better performance than existing methods for the long-short portfolio strategy. By comparing linear modeling and the proposed framework, we empirically demonstrate that the non-linear interactions between the fundamentals and lagged returns have positive impacts on the prediction results. Finally, we conduct comprehensive examinations from multiple perspectives, like the robustness of strategy and the long-short cutting percentage, to demonstrate the gross advantages. In a word, we empirically show that the higher return obtained by our method is not achieved by taking higher risks or picking a favorable threshold.

The potential future work can focus on incorporating more fundamental features and longer historical time-series data. It is noted that the intra-day momentum may perform differently from the monthly momentum, and the long-short equity strategies can be different in many ways, in which the non-linear relations among factors are not trivial to capture. So flexible deep neural networks may be a feasible tool to mine the mystery.

ACKNOWLEDGMENT

The authors would like to thank A. Nair for proofreading and all reviewers and the editor for their constructive comments, which have greatly helped to improve the presentation of the article.

REFERENCES

[1] T. Bollerslev, J. Monefar, L. Xu, and H. Zhou, “Stock return predictability and variance risk premia: Statistical inference and international evidence,” *J. Financial Quant. Anal.*, vol. 49, no. 3, pp. 633–661, Jun. 2014.
[2] I. H. Kim, A. Shamsuddin, and K.-P. Lim, “Stock return predictability and the adaptive markets hypothesis: Evidence from century-long us data,” *J. Empirical Finance*, vol. 18, no. 5, pp. 868–879, 2011.
[3] D. H. B. Phan, S. S. Sharma, and P. K. Narayan, “Stock return forecasting: Some new evidence,” *Int. Rev. Financial Anal.*, vol. 40, pp. 38–51, Jul. 2015.
[4] N. Jegadeesh and S. Titman, “Profitability of momentum strategies: An evaluation of alternative explanations,” *J. Finance*, vol. 56, no. 2, pp. 699–720, Apr. 2001.
[5] S. Gu, B. Kelly, and D. Xiu, “Empirical asset pricing via machine learning,” National Bureau of Economic Research, Cambridge, MA, USA, Tech. Rep., 2018.
[6] D. Rapach and G. Zhou, “Forecasting stock returns,” in *Handbook Economic Forecasting*, vol. 2, Amsterdam, The Netherlands: Elsevier, 2013, pp. 328–383.
[7] B. Graham, *The Intelligent Investor*. Prabhat Prakashan, New Delhi, India, 1965.
[8] R. W. Banz, “The relationship between return and market value of common stocks,” *J. Financial Econ.*, vol. 9, no. 1, pp. 3–18, Mar. 1981.
[9] S. Basu, “Investment performance of common stocks in relation to their price-earnings ratios: A test of the efficient market hypothesis,” *J. Finance*, vol. 32, no. 3, pp. 663–682, Jun. 1977.
[10] Y. Amihud, “Illiquidity and stock returns: Cross-section and time-series effects,” *J. Financial Markets*, vol. 5, no. 1, pp. 31–56, Jan. 2002.
[11] N. Jegadeesh, “Evidence of predictable behavior of security returns,” *J. Finance*, vol. 45, no. 3, pp. 881–898, Jul. 1990.
[12] L. K. Chan, N. Jegadeesh, and J. Lakonishok, “Momentum strategies,” *J. Finance*, vol. 51, no. 5, pp. 1681–1713, 1996.
[13] S. B. Marshall, “The interaction of value and momentum strategies,” *Financial Analysts J.*, vol. 27, no. 4, pp. 38–40, Nov. 1997.
[14] M. M. Carhart, “On persistence in mutual fund performance,” *J. Finance*, vol. 52, no. 1, pp. 57–82, 1997.
[15] Y. Han, G. Zhou, and Y. Zhu, “A trend factor: Any economic gains from using information over investment horizons?” *J. Financial Econ.*, vol. 122, no. 2, pp. 352–375, Nov. 2016.
[16] W. Huang, Y. Nakamori, and S.-Y. Wang, “Forecasting stock market movement direction with support vector machine,” *Comput. Oper. Res.*, vol. 32, no. 10, pp. 2513–2522, Oct. 2005.
[17] Y. Lin, H. Guo, and J. Hu, “An SVM-based approach for stock market trend prediction,” in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2015.
[18] L. Khaidem, S. Saha, and S. Roy Dey, “Predicting the direction of stock market prices using random forest,” 2016, arXiv:1605.00003. [Online]. Available: http://arxiv.org/abs/1605.00003
[19] Y. Chen and Y. Hao, “A feature weighted support vector machine and K-nearest neighbor algorithm for stock market indices prediction,” *Expert Syst. Appl.*, vol. 80, pp. 340–355, Sep. 2017.
[20] Y. LeCun and Y. Bengio, “Convolutional networks for images, speech, and time series,” in *The Handbook of Brain Theory and Neural Networks*, vol. 3, no. 10, Cambridge, MA, USA: MIT Press, 1995, p. 1995.
[21] Q. Cao, K. B. Leggio, and M. J. Schniederjans, “A comparison between Fama and French’s model and artificial neural networks in predicting the Chinese stock market,” *Comput. Operations Res.*, vol. 32, no. 10, pp. 2499–2512, Oct. 2005.
[22] D. Olson and C. Mossman, “Neural network forecasts of Canadian stock returns using accounting ratios,” Int. J. Forecasting, vol. 19, no. 3, pp. 453–465, Jul. 2003.

[23] M. Abe and H. Nakayama, “Deep learning for forecasting stock returns in the cross-section,” in Proc. Pacific–Asia Conf. Knowl. Discovery Data Mining, 2018, pp. 273–284.

[24] C. Krauss, X. A. Do, and N. Huck, “Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500,” Eur. J. Oper. Res., vol. 259, no. 2, pp. 689–702, Jun. 2017.

[25] F. Z. Xing, E. Cambria, and R. E. Welsch, “Growing semantic vines for robust asset allocation,” Knowl.-Based Syst., vol. 165, pp. 297–305, Feb. 2019.

[26] J. Li, H. Bu, and J. Wu, “Sentiment-aware stock market prediction: A deep learning method,” in Proc. Int. Conf. Service Syst. Service Manage., Jun. 2017, pp. 1–6.

[27] T. Fischer and C. Krauss, “Deep learning with long short-term memory networks for financial market predictions,” Eur. J. Oper. Res., vol. 270, no. 2, pp. 654–669, Oct. 2018.

[28] D. M. Q. Nelson, A. C. M. Pereira, and R. A. De Oliveira, “Stock market’s price movement prediction with LSTM neural networks,” in Proc. Int. Joint Conf. Neural Netw. (IJCNN), May 2017, pp. 1419–1426.

[29] J.-Z. Wang, J.-J. Wang, Z.-G. Zhang, and S.-P. Guo, “Forecasting stock indices with back propagation neural network,” Expert Syst. Appl., vol. 38, no. 11, pp. 14346–14355, 2011.

[30] M. Khasei and M. Bijari, “An artificial neural network (p.d.q) model for timeseries forecasting,” Expert Syst. Appl., vol. 37, no. 1, pp. 479–489, Jan. 2010.

[31] J.-J. Wang, J.-Z. Wang, Z.-G. Zhang, and S.-P. Guo, “Stock index forecasting based on a hybrid model,” Omega, vol. 40, no. 6, pp. 758–766, Dec. 2012.

[32] T.-J. Hsieh, H.-F. Hsiao, and W.-C. Yeh, “Forecasting stock markets using wavelet transforms and recurrent neural networks: An integrated system based on artificial bee colony algorithm,” Appl. Soft Comput., vol. 11, no. 2, pp. 2510–2525, Mar. 2011.

[33] Y.-K. Kwon and B.-R. Moon, “A hybrid neurogenetic approach for stock forecasting,” IEEE Trans. Neural Netw., vol. 18, no. 3, pp. 851–864, May 2007.

[34] A. P. Ratto, S. Merello, L. Oneto, Y. Ma, L. Malandri, and E. Cambria, “Ensemble of technical analysis and machine learning for market trend prediction,” in Proc. IEEE Symp. Ser. Comput. Intell. (SSCI), Nov. 2018, pp. 2090–2096.

[35] R. De A. Araujo, A. R. L. Junior, and T. A. E. Ferreira, “A quantum-inspired intelligent hybrid method for stock market forecasting,” in Proc. IEEE Congr. Evol. Comput., IEEE World Congr. Comput. Intell., Jun. 2008, pp. 1348–1355.

[36] S. B. Wankhade, D. Surana, N. J. Mansatta, and K. Shah, “Hybrid model based on unification of technical analysis and sentiment analysis for stock price prediction,” Int. J. Comput. Technol., vol. 11, no. 9, pp. 3025–3033, May 2018.

[37] G. Zhang, “Time series forecasting using a hybrid ARIMA and neural network model,” Neurocomputing, vol. 50, pp. 159–175, Jan. 2003.

[38] C. Cheng, F. Tan, X. Hou, and Z. Wei, “Success prediction on crowdfunding with multimodal deep learning,” in Proc. 25th Int. Joint Conf. Artif. Intell., Aug. 2019, pp. 10–16.

[39] F. Tan, Z. Wei, J. He, X. Wu, B. Peng, H. Liu, and Z. Yan, “A blended deep learning approach for predicting user intended actions,” in Proc. IEEE Int. Conf. Data Mining (ICDM), Nov. 2018, pp. 487–496.

[40] E. F. Fama and K. R. French, “The cross-section of expected stock returns,” J. Finance, vol. 47, no. 2, pp. 427–465, 1992.

[41] V. Nair and G. E. Hinton, “Rectified linear units improve restricted boltzmann machines,” in Proc. 27th Int. Conf. Mach. Learn. (ICML), 2010, pp. 807–814.

[42] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” J. Mach. Learn. Res., vol. 15, no. 1, pp. 1929–1958, 2014.

[43] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014, arXiv:1412.6980. [Online]. Available: http://arxiv.org/abs/1412.6980

[44] C. Cortes and V. Vapnik, “Support-vector networks,” Mach. Learn., vol. 20, no. 3, pp. 273–297, 1995.

[45] I. H. Nathan, “On-line support vector machine regression,” in Proc. Eur. Conf. Mach. Learn., 2002, pp. 282–294.

[46] C.-C. Chang and C.-J. Lin, “LIBSVM: A library for support vector machines,” TIST ACM Trans. Intell. Syst. Technol., vol. 2, no. 3, pp. 1–27, Apr. 2011.

[47] A. Liaw and M. Wiener, “Classification and regression by randomforest,” R News, vol. 2, no. 3, pp. 18–22, 2002.

[48] L. Breiman, “Random forests,” Mach. Learn., vol. 45, no. 1, pp. 5–32, 2001.

[49] Y. Zhai and M. Shah, “Visual attention detection in video sequences using spatiotemporal cues,” in Proc. 14th Annu. ACM Int. Conf. Multimedia (MULTIMEDIA), 2006, pp. 815–824.

[50] X. Guo, J. Zhang, Z. Wei, and H. Hakonarson, “DeepPolyA: A convolutional neural network approach for polyadenylation site prediction,” IEEE Access, vol. 6, pp. 23430–23439, 2018.

[51] K. Simonyan, A. Vedaldi, and A. Zisserman, “Deep inside convolutional networks: Visualising image classification models and saliency maps,” 2013, arXiv:1312.6034. [Online]. Available: http://arxiv.org/abs/1312.6034

XU RUI HOU received the B.E. degree in the Internet of Things from Central South University, Changsha, China, in 2015. She is currently pursuing the Ph.D. degree with the Computer Science Department, New Jersey Institute of Technology, Newark, NJ, USA. Her research interests include data mining, deep learning, and computational finance.

KAI WANG received the B.E. degree in the Internet of Things from Central South University, Changsha, China, in 2015, and the M.S. degree in financial engineering from the Stevens Institute of Technology, Hoboken, NJ, USA, in 2017, where he is currently pursuing the Ph.D. degree in finance. His research interests include asset pricing, market microstructure, and machine learning.

JIE ZHANG (Student Member, IEEE) received the B.E. degree in software engineering from Nanjing University, Nanjing, China, in 2012, and the Ph.D. degree in computer science from the New Jersey Institute of Technology, Newark, NJ, USA, in 2017. He is currently a Senior Machine Learning Research Scientist with Adobe. His research interests include data mining, bioinformatics, statistical modeling, and machine learning.

ZHI WEI (Senior Member, IEEE) received the B.S. degree from Wuhan University, China, and the Ph.D. degree from the University of Pennsylvania, USA, in 2008. He is currently a Professor of computer science and statistics (Joint appointment) with the New Jersey Institute of Technology. He has authored or coauthored more than 150 publications with 8000 citations and H-index of 38. His research interests include statistical modeling, machine learning, and big data analytics. He has served as a PC Member for the IEEE ICDM, ACM SIGKDD, the IEEE BigData, AAAI, and CIKM. He is an Editorial Board Member of BMC Genomics, BMC Bioinformatics, PLOS ONE, and the IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS.

* * *