Journal of Computer Science

Original Research Paper

Financial Forecasting with Machine Learning: Price Vs Return

[1*]Firuz Kamalov, [2]Ikhlaas Gurrib and [3]Khairan Rajab

[1]*Faculty of Engineering, Canadian University Dubai, UAE*
[2]*Faculty of Management, Canadian University Dubai, UAE*
[3]*College of Computer Science and Information System, Najran University, Saudi Arabia*

Article history
Received: 28-12-2020
Revised: 03-03-2021
Accepted: 18-03-2021

Corresponding Author:
Firuz Kamalov
Faculty of Engineering,
Canadian University Dubai,
UAE
Email: firuz@cud.ac.ae

Abstract: Forecasting directional movement of stock price using machine learning tools has attracted a considerable amount of research. Two of the most common input features in a directional forecasting model are stock price and return. The choice between the former and the latter variables is often subjective. In this study, we compare the effectiveness of stock price and return as input features in directional forecasting models. We perform an extensive comparison of the two input features using 10-year historical data of ten large cap US companies. We employ four popular classification algorithms as the basis of the forecasting models used in our study. The results show that stock price is a more effective standalone input feature than return. The effectiveness of stock price and return equalize when we add technical indicators to the input feature set. We conclude that price is generally a more potent input feature than return value in predicting the direction of price movement. Our results should aid researchers and practitioners interested in applying machine learning models to stock price forecasting.

Keywords: Stock Price Forecasting, Neural Networks, CNN, LSTM, RSI

Introduction

Stock price prediction is a classical problem in finance. It has attracted a significant amount of attention from various fields including the machine learning community. Popular economic theory in the form of Efficient Market Hypothesis postulates that financial markets are unpredictable as all the publicly available information is already incorporated in the current asset prices (Fama, 1970; Fama and French, 1988). On the other hand, the proponents of behavioral economics argue that irrational human behavior can create opportunities for arbitrage. The ability to correctly anticipate the future asset prices has enormous investment implications. As a result, a significant amount of research has been aimed at this topic. The recent advances in machine learning have provided new tools for stock market forecasting. Most notably, recurrent neural networks such as Long Short-Term Memory (LSTM) models have the ability to process sequential data such as stock prices. Despite the great promise of machine learning to uncover the hidden patterns that govern stock price movements there remains a number of gaps in our understanding of machine learning as applied to financial forecasting. One of the main issues in constructing a forecasting model is the choice of input features. Researchers use stock prices, returns, technical indicators, news and other variables as inputs. However, there is often little justification for the choice of particular features used in a forecasting model. Our goal in this study is to compare the effectiveness of two primary input features - stock price and return - in forecasting the direction of stock price movement.

Stock prediction tasks can be divided into two major categories: Price prediction and directional movement prediction. Although related, the two tasks are fundamentally different in nature. Price prediction is a regression task whereas directional movement prediction is a classification task. In this study, we study directional prediction. It is an actively researched field that is of interest to both researchers and practitioners (Borovkova and Tsiamas, 2019; Fischer and Krauss, 2018; Liew and Mayster, 2017; Kamalov *et al*., 2020a; 2020b). In particular, we are interested in the input features used in directional prediction. There are two main types of input features in most of the existing forecasting models. The first type of feature is the historical stock price. Prior stock prices are the most natural predictors of the future price movement. The existing models use stock price from one or several time steps prior to forecast the directional movement in the next time step (Borovkova and Tsiamas, 2019). The second type of input feature is the stock return. The argument for using returns states that human traders often perceive price changes in percentage terms. Stock returns of different types and time horizons are used to predict directional movement (Fischer and Krauss, 2018). We note that in addition to stock price and returns many models use a range of technical indicators as input features.

Despite the wide use of stock price and returns as input features in forecasting models the choice input features is often subjective. Both types of features are justified in the context of directional movement prediction. Since stock returns are directly related to stock price, it is often assumed that the two are interchangeable. However, it can be easily shown that if price direction movements are governed by raw prices then forecasting models that use returns as input features do not perform well (Fig. 4). Conversely, if price direction movement is determined by prior returns then forecasting models based on price do not perform well. Furthermore, the distribution of stock prices is not the same as the distribution of stock returns. Whereas the latter has always a near normal distribution the former can vary depending on a particular stock Fig. 1. Since the distribution of the features affects the optimization process of the underlying cost function, classifiers based on stock price may differ from classifiers based on returns.

The goal of our paper is to study and contrast the effectiveness of the two sets of features in price direction prediction. To this end, we carry out an extensive analysis using 10-year data on 10 large cap US stocks. The companies are chosen to represent a wide cross-section of the economy. Accordingly, the results of the study are representative of the overall stock market. In addition, we employ synthetic data to illustrate the deficiency of using an incorrect feature set. In our experiments, we utilize four standard machine learning classifiers: Logistic regression, random forest, multilayer perception and LSTM. The four algorithms represent different approaches to classification which leads to better generalization of the experimental results. The effectiveness of stock price and return values are compared in two different scenarios. First, stock price and return values are used individually as inputs to the forecasting models. Second, the two features are tested in conjunction with technical indicators. As the performance metric for classifiers we employ area under receiver operating characteristic curve. The results of numerical experiments show that price is a more effective standalone input feature than return. The results are consistent across the four classification algorithms. The difference between the two features is particularly evident when using the LSTM algorithm. The results are mixed when technical indicators are added to the models.

The paper is structured as follows. In section 2, we review the current literature on directional price movement prediction. In section 3, we describe the classifiers used in the study. In section 4, we present the results of numerical experiments. We end with concluding remarks in section 5.

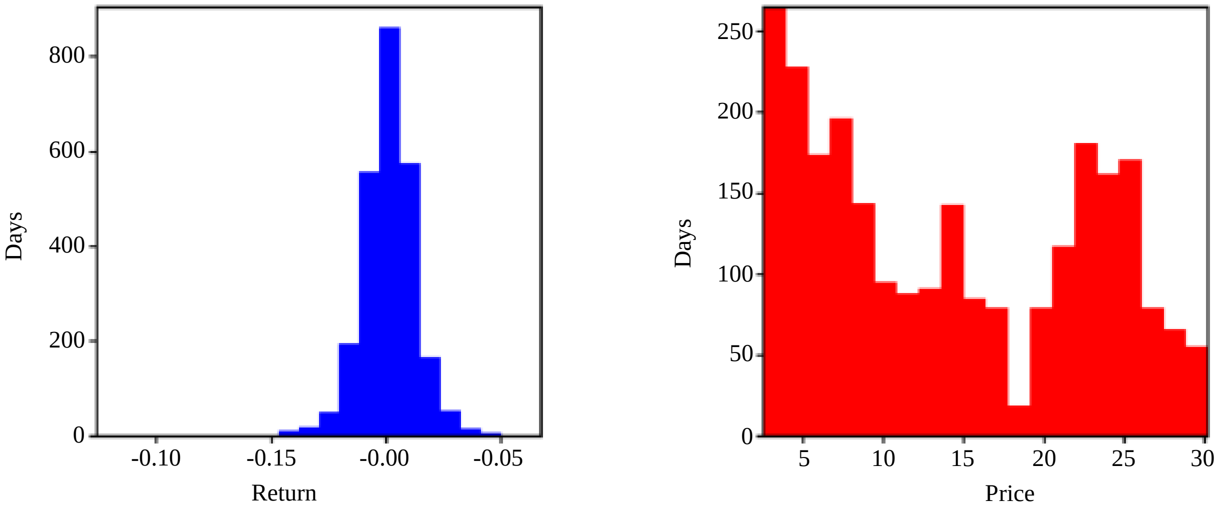


Fig. 1: Distribution of daily stock price and return values for Nike Inc. The price distribution is irregular while the return distribution is bell-shaped

Literature

There exists a large amount of literature devoted to financial forecasting. Most of the current research efforts are directed towards price forecasting and directional movement forecasting. Forecasting models that deal with directional movement employ a variety machine learning methods and input features. The input features used in the existing models can be broadly divided into four groups: Stock price, return, technical indicators and news. Stock price and return are the most widely used inputs and are the focus our study.

Stock price is used in a number of forecasting models. It is used alone as well as in conjunction with technical indicators and news. Borovkova and Tsiamas (2019), the authors applied an ensemble of LSTM models to predict directional movements for 22 large cap US stocks. The authors used high-frequency 1-year historical data. Concretely, the input features consisted of basic price based variables such as Open, Close, High, Low and others as well as more advanced technical indicators such as RSI. The proposed model is found to perform better than lasso and ridge logistic regression models. Patel *et al*. (2015), the authors apply ANN, SVM, random forest and naive Bayes classifiers to predict directional movement in Indian stock market. The authors use price based indicators to generate ten technical parameters input features. The results indicate that RF outperforms the other tested methods. Pyo *et al*. (2017) use ANN and SVM to predict the trend of the Korea Stock Price Index. The authors use price based indicators such as moving average as input features for classification. The authors obtain mixed results that are not consistent with previous research. Kamalov (2020), authors investigate the efficacy of various neural network designs such as CNN and LSTM in forecasting significant significant changes in stock price. The authors conclude that LSTMs can be used to successfully anticipate significant changes in share price for several large cap publicly traded companies. Li *et al*. (2016) applied extreme machine learning to forecast trading signal in H-share market. The authors used the intra-day tick-by-tick data and news archives for their analysis. The results have shown that the proposed method achieves both high accuracy and fast prediction speed compared to other benchmark methods. Nti *et al*. (2020a), the authors propose a fusion forecasting model based on a combination of Support Vector Machine and Genetic Algorithms (GASVM). The proposed model is used to forecast 10-day-ahead stock price movement of the Ghana Stock Exchange. The results show that the proposed method outperforms the standard classifiers including random forest, decision tree and neural network. Nti *et al*. (2020b) show that ensemble techniques can achieve robust performance. In particular, stacking and blending ensemble techniques offer higher prediction accuracy.

Stock returns have also been used in directional price movement prediction. Fischer and Krauss (2018), the authors used LSTM to predict directional movements for the constituent stocks of the S&P 500 market index. The authors use sequences of one-day returns as inputs for the LSTM model. The length of each input sequence is 240 corresponding to the daily returns over 240 days prior to the forecast date. The proposed method is found to outperform random forest, deep neural net and logistic regression models. Liew and Mayster (2017), the authors study the effectiveness of three feature subsets - returns, volume and days - on the performance of directional forecasting models. The analysis is done on a 5-year ETF data using DNNs, RFs and SVMs. The authors discover that volume is an important factor in forecasting. Kamalov and Gurrib (2020), the authors use a combination of principal component analysis and kernel density estimate to forecast significant returns in foreign exchange markets. The former technique is applied to reduce the dimension of the search space whereas the latter is employed to estimate the underlying distribution of returns. A more traditional approach to forecasting using adjusted relative strength index was proposed in (Gurrib and Kamalov, 2019). The authors modify the original relative strength index using machine learning methods to achieve better forecasting results.

Despite the wide range of forecasting models little research has been done regarding the input features. Qiu and Song (2016) study two sets of input variables for an ANN model used for stock market prediction. The two feature sets were created by the authors based on their review of the literature. The first set of features consists of technical indicators including RSI and Momentum that are loosely based on the closing price. The second group of features is comprised of technical indicators that are loosely based on return. The authors compare the effectiveness of the two sets of features in the context of predicting the direction of Nikkei 225 market index. The results show that the second group of input variables can generate a higher forecast accuracy.

One of the issues that arise when analyzing stock data is uneven distribution of price changes. If the stock data is obtained from a period of economic growth than the number of days with positive price change would outnumber the number of days with negative price change. The skewed distribution of price changes can have a negative effect on the performance of the classification algorithms (Thabtah *et al*., 2020). A common approach to address class imbalance is to balance the data through resampling (Kamalov and Denisov, 2020).

Machine Learning Models

In this section, we present a brief background on the machine learning algorithms used in our paper. We employ four popular classification algorithms in our experiments:

Logistic regression, random forest, multilayer perceptron and long short-term memory (Ballings *et al.*, 2015; Borovkova and Tsiamas, 2019; Kamalov, 2020; Patel *et al.*, 2015; Wang and Wang, 2017). The four algorithms represent different approaches to classification. Thus, we obtain a more complete analysis of the research question.

Logistic Regression (LR) is a simple linear classifier given by the equation:

$$\hat{y} = \frac{1}{1+e^{-w^T x}}, \tag{1}$$

where, $\hat{y}$ is the predicted value, w is the vector of model weights and x is the vector of features. LR has a convex cost function which ensures that a unique global minimum exists. Since LR is a linear classifier it is robust to overfitting. On the other hand, LR cannot fit nonlinear patterns. One way to address this issue is by generating nonlinear features from the original ones. Although there exist more advanced classification algorithms LR remains a standard benchmark method.

Random Forest (RF) is an ensemble estimator that fits a number of decision tree classifiers on various subsamples of the dataset. The subsample size is the same as the original input dataset with the subsamples drawn with replacement from the original data. The RF output is determined by taking the mode of decision trees used in the ensemble. In this study, we use an RF with 100 trees in the forest. The major benefit of RF is its simplicity and efficiency. In addition, the averaging procedure helps to reduce overfitting. RF is a computationally fast algorithm and serves as an excellent off-the-shelf classifier.

Multilayer Perceptron (MLP) is a classifier that is inspired by the neural architecture of the human brain. MLP and its variants have recently achieved spectacular success in image and speech recognition (Szegedy *et al.*, 2016) which prompted their use in financial modeling (Wang and Wang, 2017). The MLP architecture consists of three layers: The input layer, the hidden layer and the output layer (Fig. 2). The number of hidden layers can range from 1 to several thousand. The universal approximation theorem states that given a continuous function on a compact subset of Rn there exists an MLP with a single hidden layer and a finite number of nodes that approximates the function with any desired accuracy. Thus, MLP is a powerful model that can approximate arbitrary patterns. However, excessive fitting may lead to high variance. One way to avoid overfitting is to feed a large amount of data to the MLP model. In deep MLP models, where the number of parameters is greater than the training examples, explicit regularization is used to address overfitting. In our study, we employ an MLP model with two hidden layers with 64 and 32 nodes in the first and second layers respectively. Thus, the model is unlikely to overfit. In addition, implicit regularization that is built in the stochastic gradient descent moderates model overfitting (Arpit *et al.*, 2017).

Long Short-Term Memory (LSTM) models are Recurrent Neural Networks (RNN) that is designed to process sequential data. The classical neural network models are unable to take advantage of the chronological structure in sequential data. RNNs use a special network structure where the output at each time step depends both on the input at that time step and the state in the previous time step. However, regular RNNs are susceptible to the vanishing gradient problem where the value of the gradient decreases exponentially as it is propagated back through the network. LSTMs were proposed to solve the issue of vanishing gradients. LSTM employs special gates which control the flow of the gradient through the network in a way that maintains the gradient signal. The key concepts of an LSTM cell are cell state, forget gate, input gate and output gate. The cell state passes the information through the network connecting distant cells.

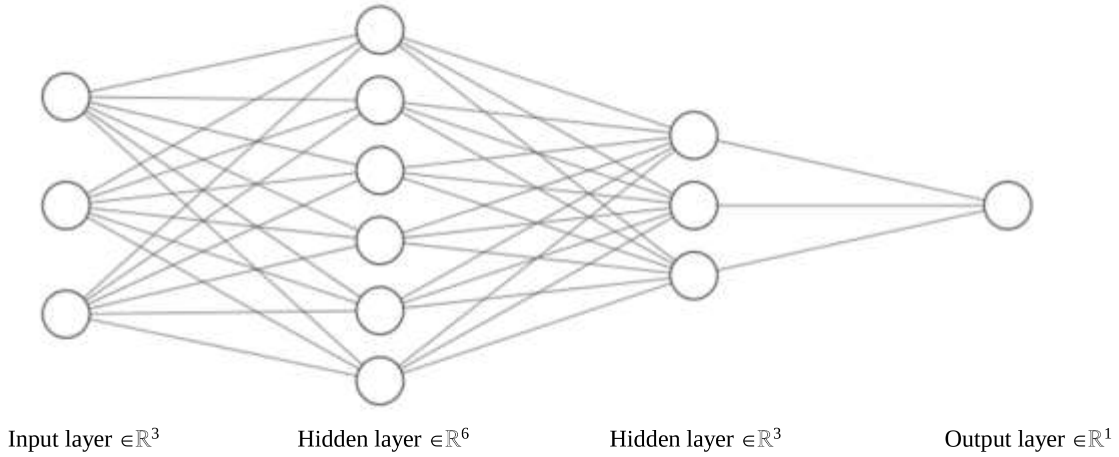

Input layer $\in\mathbb{R}^3$
Hidden layer $\in\mathbb{R}^6$
Hidden layer $\in\mathbb{R}^3$
Output layer $\in\mathbb{R}^1$

Fig. 2: Multi-layer perceptron architecture

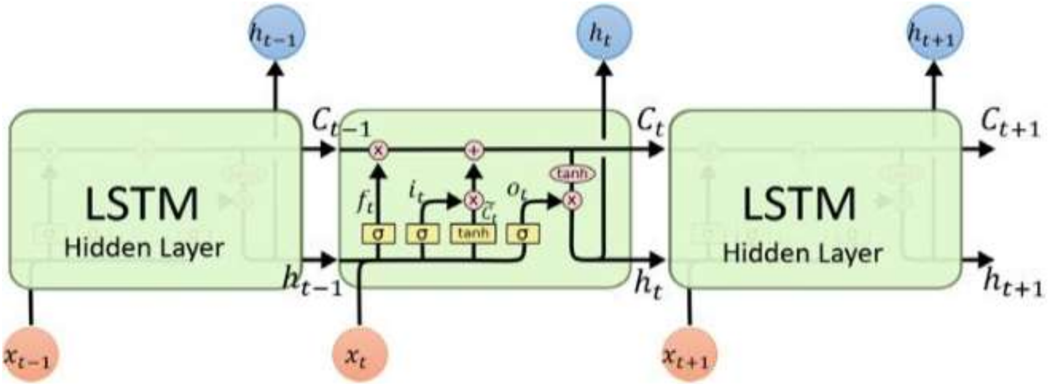


Fig. 3: The flow of information within an LSTM cell

The three gates control the flow of information inside the cell as shown in Fig. 3. The forget gate regulates the amount of information that remains in the cell. The input gate controls the amount of new information flowing into the cell and the output gate determines the next hidden state. Since LSTMs include a large number of parameters they are more likely to overfit to the training data. We employ dropout layers in the network to avoid overfitting. Dropout layers randomize the model training and reduce overfitting.

Numerical Experiments

In this section, we discuss the results of the numerical experiments that were carried out to compare the effectiveness of stock price and return. To this end, we test a range of forecasting classifiers on 10 large cap US stocks. We measure the forecasting accuracy of the models with input features consisting disjointly of prior stock price and return. In addition, we test the forecasting models using expanded feature sets that include various financial metrics. The results indicate that stock price is generally a better standalone predictor of stock price movement direction. We note that the performance of stock price and return are comparable when complemented by input features based on financial metrics.

Methodology

All the numerical experiments are carried out in Python 3.5 using the standard machine learning libraries. We use the scikit-learn library (Pedregosa *et al*., 2011) for the LR and RF classifiers. The RF model is based on 100 estimators. The neural network models MLP and LSTM are constructed using the Keras package (Chollet, 2018). The MLP model used in the experiments consists of 3 fully connected layers, where the first and second hidden layers consist of 64 and 32 nodes respectively (Fig. 2). We used ReLU activation for the hidden layers and sigmoid activation for the output layer. The RMSProp optimizer and binary crossentropy loss function were used to compile the model. The LSTM model used in the experiments has the same architecture as the MLP model. The first hidden layer is an LSTM layer with 64 nodes that returns sequence outputs. The second hidden layer is an LSTM layer with 32 nodes that returns a single output. Dropout rate of 0.2 is used for each hidden layer. The same activation, optimizer and loss functions were used as with the MLP model.

We applied the default hyperpameter settings in Keras and scikit-learn for the machine learning models. Recall that our goal is to compare the performance of stock price vs return in forecasting models. In other words, we ask that given the same model what is the better input feature? Therefore, as long as the two input features are tested on the same models the comparison results are meaningful.

The performance evaluation of classifiers is done using accuracy and Area Under receiver operating Curve (AUC). The AUC is a popular evaluation metric that produces a balanced score between the true positive and the false positive rates (Borovkova and Tsiamas, 2019; Provost and Fawcett, 2001). It represents the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance.

The experiments are performed using data on ten major publicly traded US companies (Table 2). The stock companies selected for the experiments represent a broad cross section of the US economy including manufacturing, technology, banking automotive, healthcare, pharmaceuticals and communications. We use adjusted daily stock prices from 2009 to 2019. The depth and the breadth of the datasets allow for generalization of the reported results.

The daily return is calculated based on the following formula:

$$r_t = \ln\left(\frac{p_t}{p_{t-1}}\right), \tag{2}$$

where, r_t and p_t indicate the return and price for day t respectively. To ensure the integrity of the experiments the data is split temporally into training and testing sets using a 75/25% ratio.

Table 1: Machine learning forecasting models.

Name	Implementation	Details
Logistic Regression (LR)	Scikit-learn	Default settings
Random Forest (RF)	Scikit-learn	Default settings with 100 base estimators
Multilayer Perceptron (MLP)	Keras	Default settings with 2 hidden layers 64/32, RMSProp optimizer, ReLU activation
Long Short-Term Memory (LSTM)	Keras	Default settings with 2 hidden layers 64/32, RMSProp optimizer, ReLU activation, dropout rate 0.2

Table 2: Stock companies

	Stock name (symbol)	Sector	Exchange
1	3M Company (MMM)	Manufacturing	NYSE
2	Applied Materials, Inc. (AMAT)	Technology	Nasdaq
3	Caterpillar Inc. (CAT)	Manufacturing	NYSE
4	Citigroup Inc. (C)	Banking	NYSE
5	Ford Motor Company (F)	Automotive	NYSE
6	Intel Corporation (INTC)	Technology	Nasdaq
7	Johnson & Johnson (JNJ)	Helthcare	NYSE
8	Oracle Corporation (ORCL)	Technology	NYSE
9	Pfizer Inc. (PFE)	Pharmaceutical	NYSE
10	Verizon Communications Inc. (VZ)	Communications	NYSE

Model Specifications

The goal of the numerical experiments is to compare the effects of two different sets of input features - daily price and return - on the performance of classification models. The classification task at hand is predicting the future direction of stock price movement. To this end, we create two separate sets of features and train a range of classifiers according to the individual feature sets. We evaluate the performance of the classifiers on the two feature sets and analyze the results.

In the first approach, we forecast the stock price direction based on the previous stock prices (returns) over period of p days:

$$\hat{y}_t = F\left(x_{t-1}, x_{t-2}, ..., x_{t-p}\right), \tag{3}$$

where, y_t is the predicted direction of price movement valued as 1 or -1, F is the forecasting function which is estimated by the given classifier and x_{t-i} is the price (return) of the stock at the end of day (t-i). We provide the details of the first approach in Approach 1.

Approach 1

1. Let $\left(x_t\right)_{t=0}^{N}$ be the daily stock price of a company from Table 2.
2. Reshape the data into lookback windows $X = \left(x_{t-1}, x_{t-2}, \ldots, x_{t-p}\right)_{t=p}^{N}$, where p is the number of previous days used to make the prediction. Set the target variable as the directional change of the price by $y = \left(x_t - x_{t-1}\right)_{t=p}^{N}$.
3. Train a model M from Table 1 on 75% of the data (X, y).
4. Test the model M on the remaining 25% of the data (X, y). Calculate accuracy and AUC of the model.
5. Repeat Steps 1-4 using the corresponding stock returns $\left(r_t\right)_{t=0}^{N}$.

The second model uses technical indicators in addition to basic price (return) data to forecast price direction:

$$\hat{y}_t = F\left(x_{t-1}, x_{t-2}, ..., x_{t-p}, z_{t-1}, a_{t-1}, e_{t-1}\right), \tag{4}$$

where, $z_{t\text{-}1}$, $a_{t\text{-}1}$ and $e_{t\text{-}1}$ are the RSI index, moving average and exponential moving average values respectively at the end of day t-1. The RSI index is a widely used financial indicator that represents the momentum of the stock. It is calculated based on the ratio of the sum of gains to sum of losses over period of 14 days. The level of the RSI index above 70 is normally interpreted as a sign that the stock is overbought while the index level below 30 indicates an oversold stock. The moving average is calculated as the average price (return) over period p days. The exponential moving average is calculated similar to the regular moving average using α = 0.2. We provide the details of the second approach in Approach 2.

Approach 2

1. Let $\left(x_t\right)_{t=0}^{N}$ be the daily stock price of a company from Table 2.
2. Reshape the data into lookback windows and add RSI index, moving average and exponential moving average values in the feature set $X = \left(x_{t-1}, x_{t-2}, \ldots, x_{t-p}, z_{t-1}, a_{t-1}, e_{t-1}\right)_{t=p}^{N}$. Set the target

variable as the directional change of the price by $y = \left(x_t - x_{t-1}\right)_{t=p}^{N}$.

3. Train a model *M* from Table 1 on 75% of the data (*X*, *y*).
4. Test the model *M* on the remaining 25% of the data (*X*, *y*). Calculate accuracy and AUC of the model.
5. Repeat Steps 1-4 using the corresponding stock returns $\left(r_t\right)_{t=0}^{N}$.

Results

To illustrate the effect of using 'incorrect' input features on the performance of forecasting model we construct the following synthetic example. We generate a random sequence of 5000 hypothetical daily stock prices and calculate the corresponding daily returns. We define the input features of the forecasting model to be x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5} for each *t* in the range [5, 5000]. In other words, we assume that the price movement direction is based the previous 5 days. We conduct two sets of experiments, where x_t is first taken as stock price and then as return value on day *t*. We define the class value for the dataset as the direction of change in the moving average of the return values:

$$y_t = sign\left(\sum_{t-5}^{t-1} x_i - \sum_{t-6}^{t-2} x_i\right), \tag{5}$$

where, x_i is the stock price value on day *i*. Since the response variable is determined by the raw stock price (Equation 5) we expect classification algorithms that use price values as input parameters to produce better results than models that use return values as input features. Indeed, as shown in Fig. 4, the models where the input features are prior raw price outperform the models based on prior return values. The difference in performance is significant and exists for all four classifiers. In the case of the MLP and RF models, the price based input features outperform the return based input features by over 40%. The results indicate that it is theoretically possible to obtain a dataset, where different input features can produce extremely divergent models. Although return values are directly related to price values they do not always produce the same results. Therefore, the correct choice of input features can not be underestimated. It is important, to note that the above example is artificially generated to illustrate the theoretical difference in forecasting effectiveness of price and return values as input features. In practice, the price direction is determined by a multitude of factors and the effectiveness of features will vary case to case.

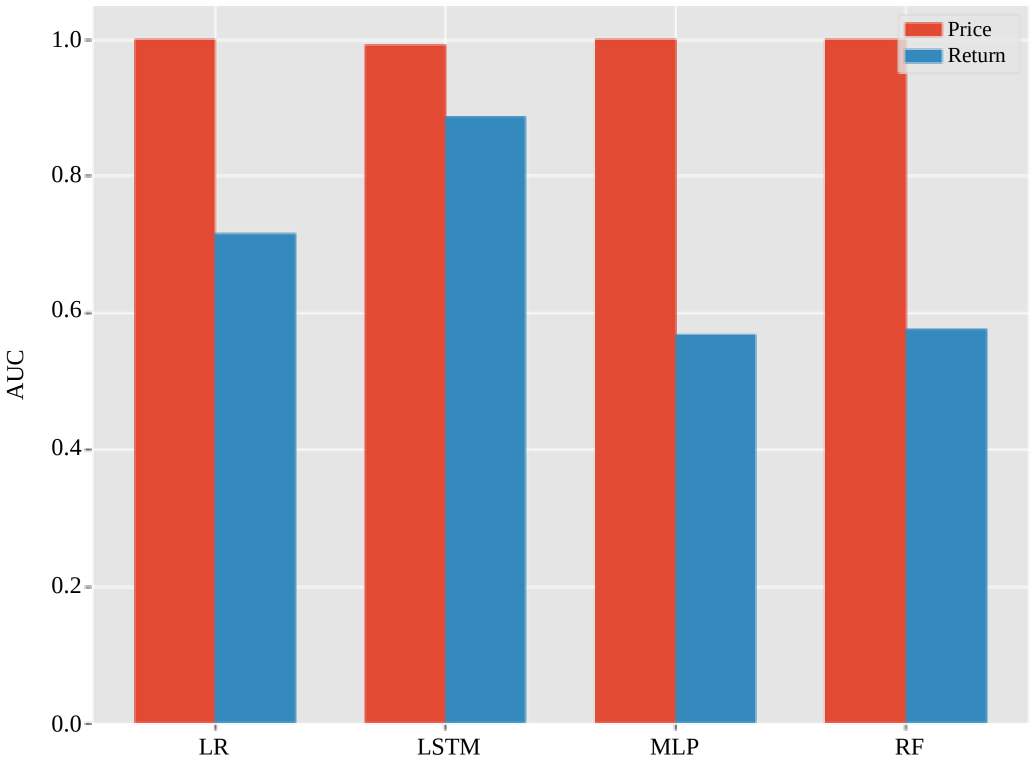
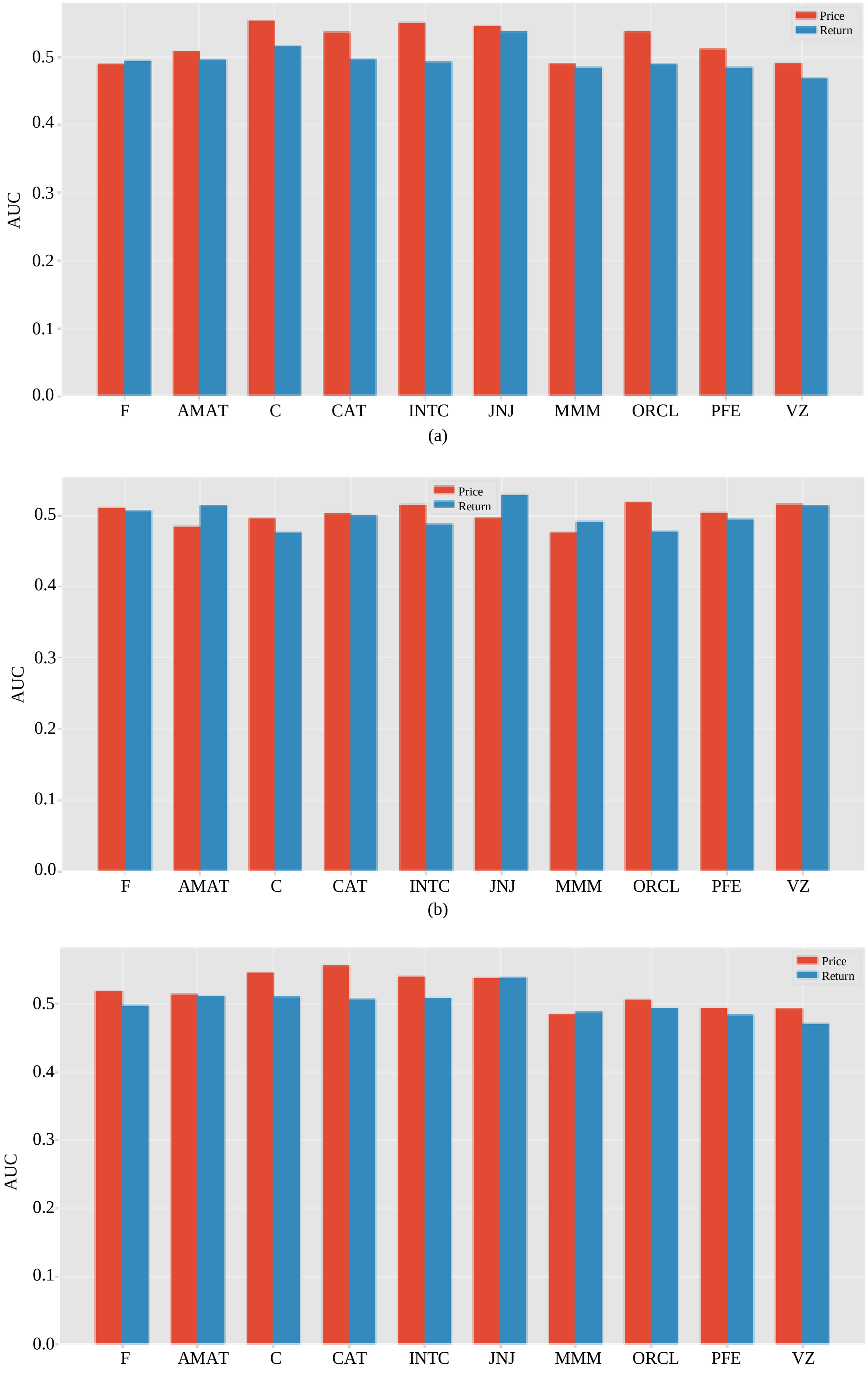


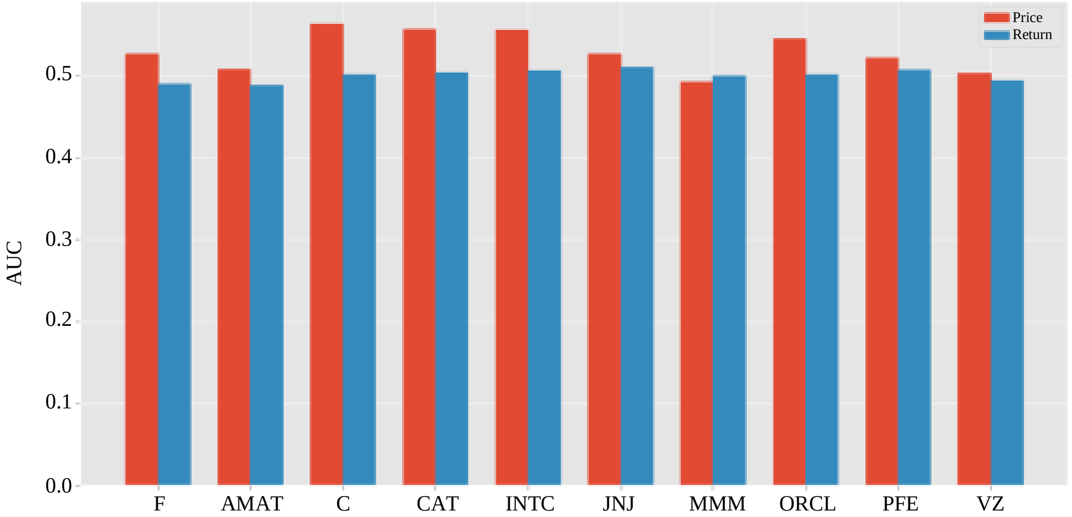
Fig. 4: Classifier performance is based on simulated input features x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, where x_t is stock price (return) on day *t*. The output is the predicted direction of price change


Price
Return
AUC
0.0
0.1
0.2
0.3
0.4
0.5
F
AMAT
C
CAT
INTC
JNJ
MMM
ORCL
PFE
VZ

(a)

(b)

(c)



(d)

Fig. 5: Experimental results as measured by average AUC. Classifier performance is based on input features x_{t-1}, x_{t-2},…,x_{t-p}, where x_t is stock price (return) on day *t* and *p* = 2, 3 and 5. The output is the predicted direction of price change; (a) Multilayer Perceptron (MLP); (b) Random Forest (RF); (c) Logistic Regression (LR); (d) Long Short-Term Memory (LSTM)

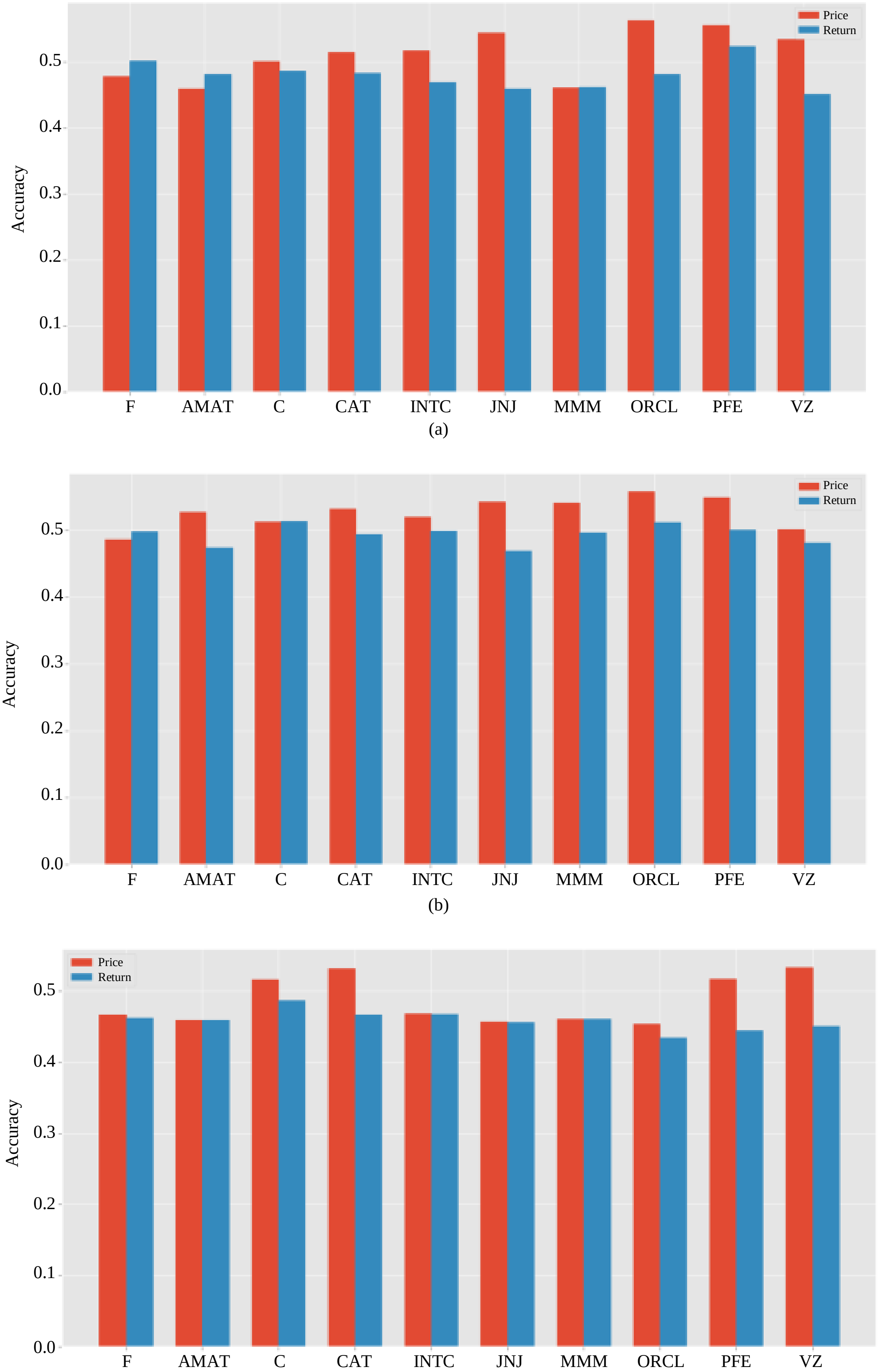
In our first experiment with real life stock data, we apply the basic model described in Equation 3. We assume that the directional movement of the stock price depends solely on the previous stock price (return). In other words, the input features of the model are prior stock price (return) over period of previous *p* days. We train four classification algorithms according to the model. The experiments are carried out for periods of *p* = 2, 3 and 5 days using each classification algorithm. We calculate the AUC for each period and report the average AUC in Fig. 5. The red and blue bars indicate the AUC values for the models trained on stock price and return values respectively.

We observe from Fig. 5 that stock price is a more effective standalone predictor of price direction movement than the return value. The advantage of stock price is consistent across different stock companies and classification models used in the experiment. The difference is particularly evident when using the LSTM algorithm (Fig. 5d). In some cases the AUC value improves by as much as 0.03 which is a significant gain in the current context. We conclude, that using prior stock prices is more effective than using return values when predicting price movement direction.

Since using stock price yields better results across different classification approaches there seems to be fundamental difference between the predictive effectiveness of stock price and return. The exact source of the difference is not quite clear and requires further research. One explanation could be the human psychology that may be more sensitive to the actual prices than returns.

To obtain the overall performance of the classifiers based on the input features we calculate the cumulative average AUC across all the stocks that are presented in Fig. 5. The results are presented in Table 3. As can be seen from Table 3, the classifiers based on the raw price have a considerably better performance than the classifiers based on the return values. The results show that all four forecasting models, on average, perform better with raw price as input feature. In particular, the MLP, LR and LSTM models, on average, perform better by 2% with raw price as input feature which is a significant improvement in the given context.

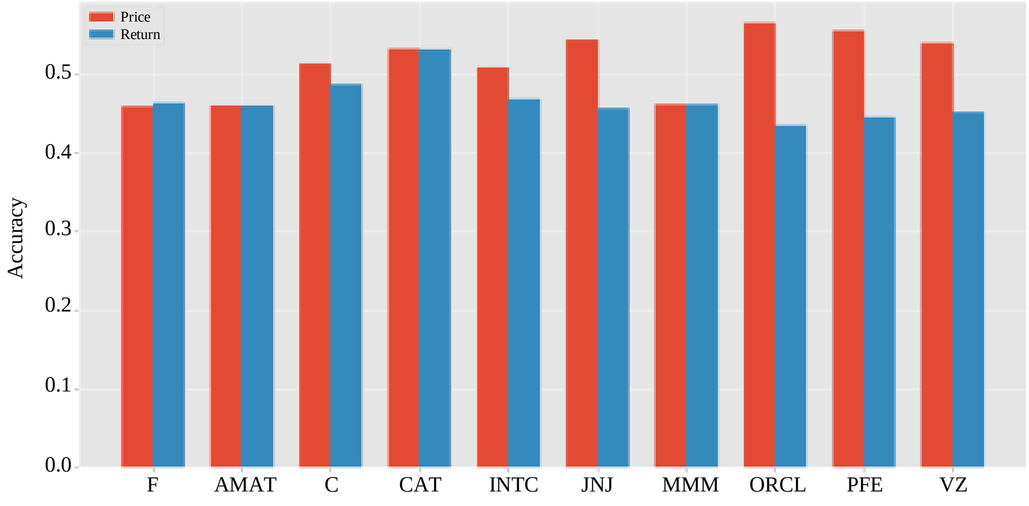
In Table 4, we present a more detailed look at the forecasting effectiveness of prior stock price and return using Intel Co stock data. We build forecasting models using stock price (return) from the previous 2, 3 and 5 days as input features. The forecasting models are constructed based on four popular classification algorithms. The experimental results, presented in Table 4, indicate that prior price is a more potent predictor of price direction than return values. The price based models outperform return based models in every tested scenario. For instance, we obtain a 6% improvement in performance when using the LSTM model with the input features consisting of the stock price over the previous 5 days. A 6% difference in performance is significant in stock price prediction (Borovkova and Tsiamas, 2019). Although the effectiveness of stock price and return varies among stocks, we generally observe a superior performance in models using stock price as a standalone predictor of price direction.


Price
Return
Accuracy
0.0
0.1
0.2
0.3
0.4
0.5
F
AMAT
C
CAT
INTC
JNJ
MMM
ORCL
PFE
VZ

(a)

(b)

(c)



(d)

Fig. 6: Experimental results as measured by average accuracy. Classifier performance is based on input features x_{t-1}, x_{t-2},…,x_{t-p}, where x_t is stock price (return) on day *t* and *p* = 2, 3 and 5. The output is the predicted direction of price change; (a) Multilayer Perceptron (MLP); (b) Random Forest (RF); (c) Logistic Regression (LR); (d) Long Short-Term Memory (LSTM)

Table 3: Average AUC results for each classifier across all the stocks presented in Figure 5.

MLPp	MLPr	RFp	RFr	LRp	LRr	LSTMp	LSTMr
0.522	0.498	0.502	0.499	0.519	0.500	0.528	0.505

Table 4: The AUC results using Intel Co stock data. Classifier performance is based on input features x_{t-1}, x_{t-2},…, x_{t-p}, where x_t is stock price (return) on day *t* and *p* = 2, 3 and 5. The output is the predicted direction of price change

Days	MLPp	MLPr	RFp	RFr	LRp	LRr	LSTMp	LSTMr
2	0.558	0.516	0.519	0.495	0.553	0.523	0.556	0.500
3	0.556	0.503	0.512	0.502	0.548	0.517	0.556	0.528
5	0.525	0.464	0.512	0.512	0.519	0.485	0.555	0.493

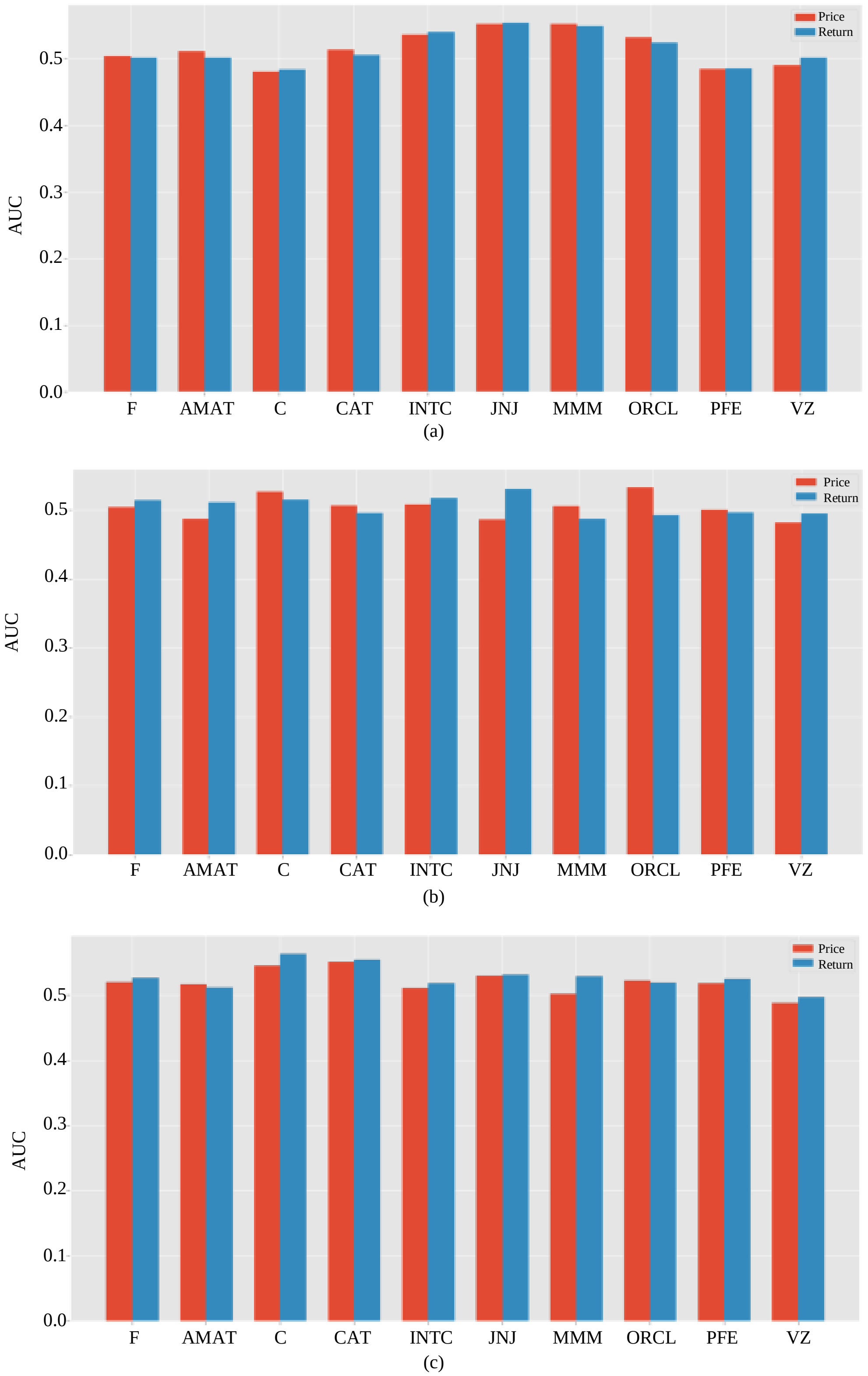
Table 5: Average accuracy results for each classifier across all the stocks presented in Figure 6.

MLPp	MLPr	RFp	RFr	LRp	LRr	LSTMp	LSTMr
0.510	0.476	0.527	0.494	0.487	0.459	0.515	0.466

The accuracy results presented in Fig. 6 further support the AUC results given above. In fact, the difference in the performance of the input features measured by accuracy is even more dramatic than measured by the AUC. For instance, when forecasting the price direction using the Oracle (ORCL) stock data based on the LSTM model the difference in accuracy of the input features is over 10%. Similar sizable differences can be observed in the case of Pfizer (PFE) and Verizon (VZ) stock data. In general, as can be seen in Fig. 6, the accuracy of stock price based classifiers is better than return based classifiers across most of the stocks tested in the experiment.

In Table 5, we present the aggregate averages across all the stocks contained in Fig. 6. As can be seen from the table, all four forecasting models have better average accuracy using raw price as input feature. The difference in performance of input features is particularly striking when using the LSTM model. The price based model outperforms the return based model by an average of 5%. Since the average values in Table 5 represent the results of extensive experiments such a difference in performance cannot be explained by pure chance. It leads us to conclude that raw stock price is a significantly better standalone predictor of price direction than return values.

In our second experiment, we include technical variables in the forecasting models. We use the RSI index, moving average and exponential moving average together with the prior stock price (return) to predict the future price direction movement. The basic model is given the Equation 4. We do not test the data with the LSTM algorithm as the new combination of features is no longer sequential. The results of numerical experiments are shown in Fig. 7. As can be seen from the figure, the predictive efficacy of stock price and return equalize with addition of the technical variables to the forecasting model. In particular, the AUC results based on using the MLP classifier show no difference between stock price and return inputs.



(a)

(b)

(c)

Fig. 7: Experimental results as measured by average AUC. Classifier performance is based on input features x_{t-1}, x_{t-2},…,x_{t-p}, z_{t-1}, a_{t-1}, e_{t-1}, where x_t is stock price (return) on day t, z_{t-1}, a_{t-1} and e_{t-1} are the RSI index, moving average and exponential moving average values respectively at the end of day t-1 and p = 2, 3 and 5. The output is the predicted direction of price change; (a) Multilayer Perceptron (MLP); (b) Random Forest (RF); (c) Logistic Regression (LR)

The results for MLP are consistent across all 10 stocks. We conclude that the addition of the technical indicators erase the advantage of price over return as a standalone feature. One possible explanation would be that the technical indicators carry the information that was lacking in the return data.

The results of our study reveal that stock price is a better standalone feature than stock return in directional forecasting. In other words, stock price is more accurate as a single input feature. However, it is important to note that stock price and return produce similar results when used in conjunction with other input variables. Our findings do not support the results of a previous study by (Qiu and Song, 2016) who found return-based technical indicators to be more effective input variables. The reason could be attributed to the difference in data employed in their study. While we used the stock price data for individual US stock companies, the authors (Qiu and Song, 2016) used data for the Japanese Nikkei 225 composite index. In the future, a study of the input features based on a larger dataset that encompasses international markets is warranted.

Conclusion

In this study, we investigated the difference between using prior stock price and return values as input features to forecast the directional movement of share price. We carried out a range of numerical experiments based on 10-year historical share price data for ten major US companies. The selected companies represent a wide cross section of the US economy. We employed four popular classification algorithms - MLP, LR, RF and LSTM - to build forecasting models. The models were run over different time horizons of p = 2, 3 and 5 days. The depth and breadth of the data together with the range of popular classifiers tested in the experiments allow us to generalize our conclusions.

Although the conventional wisdom dictates that stock price and return are closely related and thus should yield the same results, our experiments showed that it is not necessarily the case. We constructed a synthetic experiment where we showed that the forecasting efficacy can drastically differ if a ‘wrong’ feature is used (Fig. 4). The experimental results based on the real life data showed that raw stock price is a superior standalone feature than the return value (Fig. 5 and 6). The results hold across the datasets and classifiers used in the experiments. When using the LSTM classifier the price-based forecasting model produced AUC results that are on average 2.3% better than return-based model. Similarly, we observe an average of 5% difference in accuracy between price and return-based inputs. We also tested the performance of the features in the presence of technical indicators. The addition of technical variables equalized the performance of the stock and return based classifiers. It seems that the technical indicators carry the information that was lacking in return values.

It is evident from the extensive experiments that stock price is a better standalone price direction predictor than the return. However, the situation is less clear when a forecasting model contains additional technical predictors. This question requires further analysis and investigation. Another future research avenue is investigation of input features in the context of hybrid forecasting models.

The results of the experiments indicate that given a choice between raw price-based features and return-based features the former is generally more advantageous though in more complex forecasting models the difference between the two feature sets is negligible. The range of classifiers and datasets used in the experiments suggest that our findings are robust to generalization. Our results should be a useful guide to researchers and practitioners interested in applying machine learning models to stock price forecasting.

Author’s Contributions

All authors equally contributed in this work.

Ethics

This article is original and contains unpublished material. The corresponding author confirms that all of the other authors have read and approved the manuscript and no ethical issues involved.

References

Arpit, D., Jastrzebski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M. S., & Lacoste-Julien, S. (2017, August). A closer look at memorization in deep networks. In Proceedings of the 34th International Conference on Machine Learning (pp. 233-242). JMLR. http://proceedings.mlr.press/v70/arpit17a

Ballings, M., Van den Poel, D., Hespeels, N., & Gryp, R. (2015). Evaluating multiple classifiers for stock price direction prediction. Expert Systems with Applications, 42(20), 7046-7056. https://doi.org/10.1016/j.eswa.2015.05.013

Borovkova, S., & Tsiamas, I. (2019). An ensemble of LSTM neural networks for high-frequency stock market classification. Journal of Forecasting. https://doi.org/10.1002/for.2585

Chollet, F. (2018). Keras: The python deep learning library. Astrophysics Source Code Library, ascl-1806. https://ui.adsabs.harvard.edu/abs/2018ascl.soft06022C/abstract

Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. The Journal of Finance, 25(2), 383-417. https://doi.org/10.2307/2325486

Fama, E. F., & French, K. R. (1988). Permanent and temporary components of stock prices. Journal of political Economy, 96(2), 246-273. https://doi.org/10.1086/261535

Provost, F., & Fawcett, T. (2001). Robust classification for imprecise environments. Machine learning, 42(3), 203-231. https://doi.org/10.1023/A:1007601015854

Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. European Journal of Operational Research, 270(2), 654-669. https://doi.org/10.1016/j.ejor.2017.11.054

Gurrib, I., & Kamalov, F. (2019). The implementation of an adjusted relative strength index model in foreign currency and energy markets of emerging and developed economies. Macroeconomics and Finance in Emerging Market Economies, 12(2), 105-123. https://doi.org/10.1080/17520843.2019.1574852

Kamalov, F. (2020). Forecasting significant stock price changes using neural networks. Neural Computing and Applications, 32, 17655-17667. https://doi.org/10.1007/s00521-020-04942-3

Kamalov, F., & Gurrib, I. (2020). Machine learning based forecasting of significant daily returns in foreign exchange markets. arXiv preprint arXiv:2009.10065. https://arxiv.org/abs/2009.10065

Kamalov, F., & Denisov, D. (2020). Gamma distribution-based sampling for imbalanced data. Knowledge-Based Systems, 207, 106368. https://doi.org/10.1016/j.knosys.2020.106368

Kamalov, F., Smail, L., & Gurrib, I. (2020a, December). Forecasting with Deep Learning: S&P 500 index. In 2020 13th International Symposium on Computational Intelligence and Design (ISCID) (pp. 422-425). IEEE. https://doi.org/10.1109/ISCID51228.2020.00102

Kamalov, F., Smail, L., & Gurrib, I. (2020b, November). Stock price forecast with deep learning. In 2020 International Conference on Decision Aid Sciences and Application (DASA) (pp. 1098-1102). IEEE. https://doi.org/10.1109/DASA51403.2020.9317260

Li, X., Xie, H., Wang, R., Cai, Y., Cao, J., Wang, F., & Deng, X. (2016). Empirical analysis: stock market prediction via extreme learning machine. Neural Computing and Applications, 27(1), 67-78. https://doi.org/10.1007/s00521-014-1550-z

Liew, J. K. S., & Mayster, B. (2017). Forecasting etfs with machine learning algorithms. The Journal of Alternative Investments, 20(3), 58-78. https://doi.org/10.3905/jai.2018.20.3.058

Nti, I. K., Adekoya, A. F., & Weyori, B. A. (2020a). Efficient Stock-Market Prediction Using Ensemble Support Vector Machine. Open Computer Science, 10(1), 153-163. https://doi.org/10.1515/comp-2020-0199

Nti, I. K., Adekoya, A. F., & Weyori, B. A. (2020b). A comprehensive evaluation of ensemble learning for stockmarket prediction. Journal of Big Data, 7(1), 1-40. https://doi.org/10.1186/s40537-020-00299-5

Qiu, M., & Song, Y. (2016). Predicting the direction of stock market index movement using an optimized artificial neural network model. PLoS ONE, 11(5), e0155133. https://doi.org/10.1371/journal.pone.0155133

Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. Expert Systems with Applications, 42(1), 259-268. https://doi.org/10.1016/j.eswa.2014.07.040

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., & Vanderplas, J. (2011). Scikitlearn: Machine learning in Python. Journal of machine learning research, 12(Oct), 2825-2830. https://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf?source=post_page---------------------------

Pyo, S., Lee, J., Cha, M., & Jang, H. (2017). Predictability of machine learning techniques to forecast the trends of market index prices: Hypothesis testing for the Korean stock markets. PLoS ONE, 12(11), e0188107. https://doi.org/10.1371/journal.pone.0188107

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2818-2826). IEEE. https://doi.org/10.1109/CVPR.2016.308

Thabtah, F., Hammoud, S., Kamalov, F., & Gonsalves, A. (2020). Data imbalance in classification: Experimental evaluation. Information Sciences, 513, 429-441. https://doi.org/10.1016/j.ins.2019.11.004

Wang, J., & Wang, J. (2017). Forecasting stochastic neural network based on financial empirical mode decomposition. Neural Networks, 90, 8-20. https://doi.org/10.1016/j.neunet.2017.03.004