Expert Systems With Applications 168 (2021) 114444

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Forecasting daily stock trend using multi-filter feature selection and deep learning

Anwar Ul Haq, Adnan Zeb, Zhenfeng Lei, Defu Zhang *

School of Informatics, Xiamen University, Xiamen, Fujian, 361005, China

ARTICLE INFO

Keywords:
Stock trend prediction
Feature selection
Deep learning
Machine learning

ABSTRACT

Stock market forecasting has attracted significant attention mainly due to the potential monetary benefits. Predicting these markets is a challenging task due to numerous interrelated factors, and needs a complete and efficient feature selection process to identify the most informative factors. As a time series problem, stock price movements are also dependent on movements on its previous trading days. Feature selection techniques have been widely applied in stock forecasting, but existing approaches usually use a single feature selection technique, which may overlook some important assumptions about the underlying regression function linking the input and output variables. In this study, we combine features selected by multiple feature selection techniques to generate an optimal feature subset and then use a deep generative model to predict future price movements. First, we compute an extended set of forty-four technical indicators from daily stock data of eighty-eight stocks and then compute their importance by independently training logistic regression model, support vector machine and random forests. Based on a prespecified threshold, the lowest ranked features are dropped and the rest are grouped into clusters. The variable importance measure is reused to select the most important feature from each cluster to generate the final subset. The input is then fed to a deep generative model comprising of a market signal extractor and an attention mechanism. The market signal extractor recurrently decodes market movement from the latent variables to deal with stochastic nature of the stock data and the attention mechanism discriminates between predictive dependencies of different temporal auxiliary outputs. The results demonstrate that combining features selected by multiple feature selection approaches and using them as input into a deep generative model outperforms state-of-the-art approaches.

1. Introduction

Stock markets play an important role in the organization of modern economic systems and forecasting these markets is very important for investors, as accurate predictions allow them to mitigate risks and make informed decisions about investments. Stock markets are influenced by various factors such as economic situation, industry specific variables, company outlook, psychological influence of investors and government policies, adding to the complexity of analyzing and predicting these markets (Zhong & Enke, 2017). Moreover, the efficient market hypothesis (EMH) by Malkiel and Fama (1970) states that stock markets reflect all available information and follow random pathways, declaring them extremely difficult to be accurately predicted. However, many empirical studies suggest that financial markets do not immediately adjust to newly released information, and psychological influence of various market participants make financial markets predictable to some extent (Cervelló-Royo & Guijarro, 2020; Garcia et al., 2018; Henrique et al., 2019), allowing investors to make profits above market average by analyzing publicly available information. Cervelló-Royo and Guijarro (2020) employed five different machine learning models and showed that all models achieve prediction accuracy above average. Lately, deep learning approaches have appeared with results that outperforms their traditional machine learning counterparts (Sezer et al., 2020).

In general, stock forecasting approaches are categorized into fundamental analysis and technical analysis, based on the type of information each approach relies upon. In fundamental analysis, investors estimate intrinsic value of stock by examining sales, profits, debits and dividends of a company. The information used for fundamental analysis is periodically released by companies and are of little use when explaining high-frequency price moments. Fundamental analysis based models are less commonly found in the literature as it is harder to build models that understands price movements using fundamental indicators (Bustos & Pomares-Quimbaya, 2020). While, technical analysis evaluates stocks by analyzing statistical trends generated by market activity such as

* Corresponding author.
E-mail addresses: anwar@uom.edu.pk (A.U. Haq), adnanzeb@stu.xmu.edu.cn (A. Zeb), zflei621@stu.xmu.edu.cn (Z. Lei), dfzhang@xmu.edu.cn (D. Zhang).

https://doi.org/10.1016/j.eswa.2020.114444
Received 30 May 2020; Received in revised form 5 September 2020; Accepted 2 December 2020
Available online 5 December 2020
0957-4174/

historical prices and volumes. Technical analysis is commonly used for short term forecasting as the required information is regularly released.

In technical analysis, meaningful information about stocks is extracted by computing technical indicators from past prices and volumes, which are then given as input to a prediction model. Adding these indicators may result in irrelevant or redundant information, which not only increases computational cost but may also result in lower prediction performance. Identifying financial indicators with high informational value requires in depth knowledge of the field, which may not be always available. Moreover, the performance of a learning model heavily depends upon the data representation method. Therefore, transforming raw data before inputting it into a model often improves performance of a learning model.

Like other machine learning problems, stock forecasting also involves the selection of suitable features from a given set and then using them to train a prediction model. Successful stock prediction aims to achieve highest prediction accuracy with minimum input and the least complex learning model (Atsalakis & Valavanis, 2009). Feature selection enables analysts to remove irrelevant and/or redundant features from a data set to reduce dimensionality and improve prediction performance. In stock forecasting (Zhong & Enke, 2017) employed PCA, fuzzy-robust PCA and kernel-based PCA for feature selection and found that combining ANN with PCA achieves slightly higher classification accuracy than other approaches evaluated. Gündüz et al. (2017) used gain ratio and Relief for feature selection to predict stock price direction using logistic regression and gradient boosting machine. Inspired by the superior performance of ensemble learning approaches, Tsai and Hsiao (2010) combined features selected by PCA, GA and CART for stock prediction and concluded that features selected by intersection between PCA and GA have slightly better prediction accuracy and lower error.

Majority of the existing studies usually use a single feature selection technique and tend to predict a single stock index or stock price. However, some studies were also conducted on forecasting returns or price movement for multiple stocks (Cervelló-Royo & Guijarro, 2020; Picasso et al., 2019; Xu & Cohen, 2018). Existing approaches also differ in the type and number of input variables used in each study, with technical indicators being the most commonly used (Atsalakis & Valavanis, 2009). More recently, there is a growing trend of using information from news articles, blogs and social media to forecast financial variables. TSLDA (Nguyen & Shirai, 2015) used historical prices and information from Yahoo message board to predict stock movement. In HAN (Hu et al., 2018), the authors proposed application of hierarchical attention mechanism to news sequence directly mined from text. In STOCKNET (Xu & Cohen, 2018), the authors combined historical prices with tweets to predict next-day price movement.

For prediction, both statistical and soft computing models have been explored to analyze stock markets. Statistical models such as ARIMA (Autoregressive Independent Moving Average) (Brown, 2004), GARCH (Generalized Autoregressive Conditional Heteroskedasticity) (Bollerslev, 1986) and SV (Stochastic Volatility) are still used for time series forecasting in economics, mainly due to their well understood mathematical properties. However, statistical methods assume stock markets to be linear, stationary and normal, which limit their performance and practical application.

Machine learning models such as SVM, artificial neural networks (ANN), fuzzy systems and genetic algorithms have become popular in financial forecasting, as they are driven by multivariate data without any prespecified assumption (Zhong & Enke, 2017). Among these methods, SVM has been widely used for both predicting stock movements and stock prices (Huang et al., 2007; Lin et al., 2013; Trafalis & Ince, 2000). Application of SVM to predict daily price change in Korean composite stock price index (KOSPI) (Kim, 2003) is still one of the most cited work in the field. Combining multiple approaches to develop new solutions is also prevalent, as (Cai et al., 2013; Ye et al., 2016) the authors used technical analysis in combination with genetic algorithm and fuzzy systems, respectively.

The ability of ANN to mine information from the plethora of historical data and effectively use it for future forecasting has also made it a popular choice (Zhang et al., 2007). In a recent study (Henrique et al., 2019), the authors reviewed 547 articles and found that nearly 74% of the studies surveyed used at least some form of ANN for stock forecasting, followed by SVM used by 37%. In a comparative study (Guresen et al., 2011), the authors demonstrated that classical ANN outperforms a dynamic ANN (Ghiassi et al., 2005) and a hybrid model (GARCH-ANN) (Roh, 2007). Adebiyi et al. (2014) suggested that the difference between the predicted values of ANN and ARIMA is insignificant. Although, ANN seems to perform well in financial forecasting, their performance strongly depends on the optimal selection of input features and the combination of network parameters. More recently, deep generative models have become popular in image classification, text mining and translation, but have not been used in financial forecasting, the only exception being STOCKNET. STOCNET uses all the input features to train the model, where some might be irrelevant or redundant. In order to provide a more optimal feature set, we combine features selection with STOCKNET to improve its prediction performance. Unlike existing studies, which use a single feature selection technique (Gündüz et al., 2017; Lin et al., 2013; Zhong & Enke, 2019), we combine features selected by three different feature-ranking approaches having disjoint assumption about the regression function linking the input and output variables. To the best of our knowledge, this is the first attempt to study the effect of dimensionality reduction on deep generative model for stock trend prediction.

In this study, we compute an extended set of 44 technical indicators and use a multi-filter feature selection approach (Haq et al., 2019) to select an optimal feature set that can efficiently represent the original data set. The feature selection approach independently applies $L1$ regularized logistic regression ($L1$-LR), support vector machine (SVM) and random forest (RF) to identify and select an optimal feature subset. The generated feature set is then fed to a deep generative model for classifying them into up or down classes. The prediction model is based on STOCKNET (Xu & Cohen, 2018), and uses Variational AutoEncoders (VAE) (Kingma & Welling, 2013) to recurrently infer and extract latent driven factors and stock movements from the observed stock data. These auxiliary predictions have varying degree of influence on predicting the main target, therefore an attention mechanism is added to discriminate between these auxiliary predictions. The results demonstrate that combining features selected by multiple disjoint feature selection approaches improve prediction performance of STOCKNET.

The remainder of the paper is structured as follow. In Section 2, we describe the data and the pre-processing steps used to compute technical indicators. In Section 3, the feature selection techniques are described, and Section 4 provides detail about the prediction model. In Section 5, experimental setup of the feature selection module and prediction model are discussed. Section 6 compare the results with baseline approaches followed by conclusion in Section 7.

2. Data description and preprocessing

In this study, we use the data set from Xu and Cohen (2018), which contains daily price movements of 88 NASDAQ listed stocks from 01/01/2014 to 01/01/2016. The data set includes 8 stocks from *Conglomerates* and top 10 stocks by capital size from each of the 8 industries namely *Basic Materials, Utilities, Healthcare, Services, Consumer Goods, Finance, Industrial Goods and Technology*. Each stock has a daily open, low, high, close, adjusted close and volume. The total number of daily observations is 43,309.

As predictions are based on relative price change, we first normalize the input values and then compute technical indicators. Considering the raw price vector $p_t' = [OP_t', HP_t', LP_t', CP_t', AP_t']$, containing open, low, high, close and adjusted close price on trading day t. We normalize the absolute values by using the previous day adjusted close price as:

Table 1
Technical indicators used.

Indicator	Description	Indicator	Description
OP	Open price	LL(x)	Lowest price
HP	High price	HH(x)	Highest price
LP	Low price	MA(x)	Simple moving average
CP	Close price	EMA(x)	Exponential MA
MED	Median price	WMA(x)	Weighted MA
TYP	Typical price	DEMA(x)	Double EMA
MEAN	Mean price	TEMA(x)	Triple EMA
ROC(x)	Rate of change	KAMA(x)	Kaufman adaptive MA
MFI(x)	Money flow index	MID(x)	Midpoint price
MOM(x)	Momentum	MACD(x,y)	Moving Avg. Con/Div
RSI(x)	Relative strength Index	MACDS(x,y)	MACD signal
WILLR(x)	Williams' % R	MACDH(x,y)	MACD histogram
AD	Chaikin A/D Line	PLUS-DI(x)	Plus directional indicator
CO(x,y)	Chaikin A/D oscillator	PLUS-DM(x)	Plus directional movement
SK(x,y)	Slow stochastic % K	CCI(x)	Commodity channel index
SD(x,y)	Slow stochastic % D	PPO(x,y)	Percentage price oscillator
FK(x,y)	Fast stochastic % K	DX(x)	Directional movement index
FD(x,y)	Fast stochastic % D	ADX(x)	Avg. DX
UBB(x)	Upper Bollinger bands	ADXR(x)	ADX rating
LBB(x)	Lower Bollinger bands	TRANGE	True range
MBB(x)	Middle Bollinger bands	ATR(x)	Avg. true range
PCTBB(x)	Pct. Bollinger bands	NATR(x)	Normalized ATR

Note: x = 5 trading days and y = 10 trading days.

$p_t = (p'_t/AP_{t-1})-1$. Similarly, the volumes are normalized with the value of previous day volume.

Price movements with exceptionally small movement ratios are filtered-out by setting lower and upper thresholds. Since we treat stock movement prediction as a binary classification problem, we set two thresholds, −0.5% and 0.55%, and filter-out 16,695 records having movement percentages between the two thresholds. As in Xu and Cohen (2018), stock records with change percentage smaller than −0.5% and greater than 0.55% are assigned 0 and 1 labels, respectively. The two thresholds also balance the number of observations in each class, resulting in 26,614 total observations in the data set with 49.78% and 50.22% in up and down class respectively. The remaining records are split and 20,339 records between 01/01/2014 and 01/08/2015 are used for training, 2555 records from 01/08/2015 to 01/10/2015 are used for validation, and 3720 records from 01/10/2015 to 01/01/2016 are used for testing.

In order to extract maximum information, forty four financial indicators were identified from existing studies (Gündüz et al., 2017; Wu et al., 2014; Ye et al., 2016) and used in this study. Table 1, lists the indicators along with their description. For training the feature ranking techniques and the prediction model, we estimate the binary movement as the difference between a trading day adjusted close price and its previous day adjusted close price as $y = 1(AP_{t-1} > AP_t)$, where 1 denotes rise and 0 denotes fall. For prediction, We use previous day information along with other valid trading days in its lag $[t- \triangle t, \ldots, t-1]$ as input, where $\triangle t$ is a fixed lag size.

3. Preliminary concepts

3.1. Feature selection

Identifying and selecting input features with high predictive abilities has long been a research topic in data mining. Feature selection aims to filter-out irrelevant and redundant input variables to reduce complexity and improve prediction accuracy. Suppose, $\{\boldsymbol{x_n}, \boldsymbol{y_n}\}$ represents the raw data set, where $\boldsymbol{x_i}$ is a $\boldsymbol{d}$-dimensional vector and $\boldsymbol{y_i}$ is the corresponding class label. The ith instance of the data set is denoted by $x_{i1}, x_{i2}, \ldots, x_{id}, y_i$, and suppose we have n such i.i.d. observations. Feature selection aims to find $\boldsymbol{x'_i}$, a $\boldsymbol{d'}$-dimensional vector of variables, which can efficiently explain the associated class label $\boldsymbol{y_i}$, where $\boldsymbol{d'} \ll \boldsymbol{d}$. Moreover, for two variables $\{x_{ij}, x_{ik}\}$ in $\boldsymbol{x_i}$, x_{ij} is considered to have higher predictive abilities, if it explains more variability in $\boldsymbol{y_i}$ than x_{ik}.

3.2. Multi-filter feature selection

In order to combine the strengths of multiple feature-ranking techniques, we combine features selected by three different feature-ranking approaches. In Haq et al. (2019), the authors combined clustering of variables with L1–LR, SVM and RF based feature-ranking to select an optimal feature set. The selected feature-ranking approaches have better performance and are considered to have few hyper-parameters, which make them easy to optimize. Their selection is primarily based on having disjoint assumptions about the regression function linking the predicted variable with predictors. For example, L1–LR assumes the regression function to be linear, while RF considers variables which may statistically interact in their effect on the target variable. The authors compared performance of MFFS with that of PCA, ReliefF, SFFS, PSO-SVM, InfoGain and GA-based feature selection methods and found that MFFS outperforms other approaches evaluated. The three feature-ranking techniques and the clustering method are discussed below:

3.2.1. L1-LR feature ranking

L1–LR, a popular statistical classification method, has recently received increasing attention as a feature selection technique. L1–LR models the class membership probabilities as a linear combinations of input variables and build an easy to understand linear model. A number of existing studies (Ng, 2004; Zakharov & Dupont, 2011) have demonstrated it to be an effective feature selection approach. The $L1$ regularized logistic regression objective function for n observations can be written as:

$$min\ \ell_{avg}(\alpha,\beta) + \lambda|\beta|_1 = \frac{1}{n}\sum_{i=1}^{n}\log(1+\exp(-(\beta^T a_i + \alpha y_i))) + \lambda\sum_{i=1}^{n}|\beta_i| \quad (1)$$

where $a_i = x_i y_i$, the $L1$ norm $|\beta|_1 = \sum_{i=1}^{n}|\beta_i|$ and $\lambda > 0$ is the regularization parameter. β^T is the weight vector and provide insight into the predictive abilities of each feature. $L1$-LR typically produces a sparse vector β with few non zero values. Input features with regression coefficient 0 will be discarded e.g. if $\beta_j = 0$, the jth item of the input vector will be discarded. Thus, the weight vector can be used for identifying highly informative features and filtering-out those with smaller weight values.

3.2.2. SVM-based feature ranking

SVM, another well known learning algorithm, classifies samples by finding the optimal hyperplane using margin maximization. The concept has been discussed in detail in numerous studies, including (Hearst et al., 1998; Huang et al., 2007; Lee, 2009). Considering the data set $\{\boldsymbol{x_n}, \boldsymbol{y_n}\}$, SVM first transforms the input $\boldsymbol{x_i}$ into a high-dimensional space using a function $\boldsymbol{\phi}$ and then computes a decision function of the form $f(x) = sgn(w \cdot \boldsymbol{\phi}(x_i) + b)$. The objective is to maximize the margin parameterized by w and b, and the sgn of the function is used to assign class labels. The optimization problem for the above decision function with misclassified instances penalized can be written as:

$$\min_{w,b,\xi} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i \quad (2)$$

$$subject\ to \quad y_i(w \cdot x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \ldots, n$$

where C is the penalty parameter and ξ_i is the slack variable. SVM utilizes non-negative slack variables to relax the constraint and tolerate some misclassification during training. The optimization problem in Eq. (2) is solved by Lagrangian method, where the objective is to maximize over α_i.

$$\max_{\alpha}\sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{n}y_i y_j \alpha_i \alpha_j (x_i \cdot x_j), \quad s.t. \quad \sum_{i=1}^{n}\alpha_i y_i = 0 \ \ \& \ \ 0 \leq \alpha_i \leq C \quad (3)$$

The solution for α_i determines the parameters w and b for the optimal hyperplane and the resulting weight vector $w \in \mathbb{R}^d$ is used as variable importance measures (Guyon et al., 2002; Li et al., 2011; Rakotomamonjy, 2003).

3.2.3. RF-based feature ranking

Random forests (Breiman, 2001), a statistical method used for classification and regression problems, also provides useful internal estimates of error, strengths, correlations and variable importance. An RF is a collection of decision trees (*ntree*), and each tree is built by drawing random samples (with replacement) from the data set. At each node the split is also randomly selected from a number (*mtry*) of best splits. In RF, variable importance is calculated by comparing prediction errors (mean decrease in accuracy) before and after permuting the values of the concerned variable. Variable importance for a specific feature x_i within each tree t is computed as in Eq. (4).

$$VI_t(x_i) = \frac{1}{|\overline{B}_t|}\Big(\sum_{i\in\overline{B}_t} I(y_i = \tilde{y}_{it}) - \sum_{i\in\overline{B}_t} I(y_i = \tilde{y}^*_{it})\Big) \tag{4}$$

where $I(\cdot)$ is the indicator function, $\tilde{y}_{it}$ and $\tilde{y}^*_{it}$ are the predicted values before and after permuting the value of feature x_i, respectively. $\overline{B}_t$ is the set of indices of out-of-bag observations for tree $t \in \{1, \dots, ntree\}$. The raw importance of each variable is then calculated over all trees in the forest as:

$$VI_{xi} = \frac{1}{ntree}\sum_{t=1}^{ntree} VI_t(x_i) \tag{5}$$

Since, input variables are ranked differently for each stock, we calculate the mean importance of each variable to get a generic view of variable importance for all stocks. We filter-out features with lower predictive abilities than the given threshold and the rest of the features are clustered into groups using affinity propagation (Frey & Dueck, 2007). Affinity propagation, an exemplar-based clustering approach identifies data-points that best exemplify the data. It takes a function or real-valued similarities to measure correlation among data points. As a similarity measure, we use linear correlation among the remaining variables for clustering them into groups. Affinity propagation does not require a prespecified number of clusters but takes a real number input for each data point in the similarity matrix so that data points with larger values be selected as exemplars. These values are referred to as "preferences" denoted by $\boldsymbol{p}$. The input preferences along with the message passing procedure control the selection of cluster centers and the number of clusters to be produced. Algorithm 1 provides detailed operation of the feature selection approach.

Algorithm 1 Multi-Filter Feature Selection (Haq et al., 2019)

Input: $S := \{(x_1, y_1), ..., (x_n, y_n)\}$ x_i is a d-dimensional vector
Output: $S' := \{\{(x_1, y_1), ..., (x_n, y_n)\}$ x_i is a d'-dimensional vector
▷ such that $d' << d$

procedure MFFS(S, $\boldsymbol{e}, \boldsymbol{p}$)
  delegates ← ∅
  $\boldsymbol{SF} \leftarrow \emptyset$ ▷ selected features
  $xtrain, ytrain, xtest, ytest \leftarrow$ preprocess(S)
  $VI_{lr} \leftarrow \boldsymbol{L1-LR}$ $(xtrain, ytrain)$
  $VI_{svm} \leftarrow \boldsymbol{SVM}$ $(xtrain, ytrain)$
  $VI_{rf} \leftarrow \boldsymbol{RF}$ $(xtrain, ytrain)$
  for each $VI \in [VI_{lr}, VI_{svm}, VI_{rf}]$ **do**
    $xtrain \leftarrow xtrain$.drop($VI, \boldsymbol{e}$) ▷ $\boldsymbol{e}$: elimination threshold.
    clusters ← Cluster($xtrain$, $\boldsymbol{p}$) ▷ $\boldsymbol{p}$: preference parameter.
    delegates ← Delegates(VI, clusters)
    for each feature in delegates **do**
      if feature ∉ SF **then**
        SF.append(feature)
      end if
    end for
  end for
  $xtrain', xtest' \leftarrow xtrain[\boldsymbol{SF}], xtest[\boldsymbol{SF}]$
  $S' \leftarrow$ concatenate($xtrain', xtest'$)
  return S'
end procedure

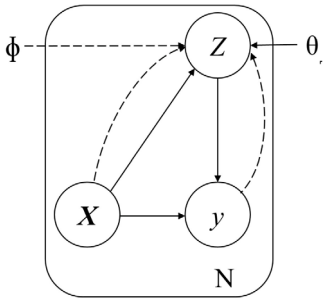

Fig. 1. Graphical representation of a generative process for N instances of observed information X to predict stock movement y through latent variable Z. Solid lines represent the generative process and dashed lines denote the variational approximation to the intractable posterior. Both generative parameters θ and variational parameters ϕ are jointly learned.

4. Model description

As stock forecasting is a time series problem, it is assumed that predicting stock movement on a certain trading day d will benefit from predicting it for its lag days (Xu & Cohen, 2018). For this purpose, we first identify D temporally close eligible trading days referred in a sample, in other words eligible trading days in a time interval $[d-\triangle d, d-1]$. Then incorporate those lag records to produce new encoded input of the form $X = \{x_1, \dots, x_D\}$ and the corresponding sequence of target predictions of the form $y = \{y_1, \dots, y_D\}$. The main prediction target is y_D, while the remaining $y^* = \{y_1, \dots, y_{D-1}\}$ are temporal auxiliary targets, which will help in improving prediction accuracy of the main target.

We assume that there exists a continuous latent variable $Z = \{z_1, \dots, z_D\}$ generated from the observed market information, and this variable along with X guide the prediction process i.e. $p_\theta(y|X, Z)$, as shown in Fig. 1. With the above assumption, the original conditional probability evolves into $p_\theta(y|X) = \int_z p_\theta(y, Z|X)dz$, and the factorization for the generative process can be written as:

$$p_\theta(y, Z|X) = p_\theta(y_D|X, Z)p_\theta(z_D|z_{<D}, X)\prod_{d=1}^{D-1} p_\theta(y_d|x_{\le d}, z_d)p_\theta \times (z_d|z_{<d}, x_{\le d}, y_d) \tag{6}$$

Since the auxiliary targets are already known in the generation; when $d < D$, we use the posterior probability $p_\theta(z_d|z_{<d}, x_{\le d}, y_d)$, to accurately incorporate market signals and use the prior probability $p_\theta(z_D|z_{<D}, X)$ when generating $z_{<D}$. Moreover, y_d is independent of $z_{<d}$ when $r_{<D}$, while the effect of $z_{<D}$ is utilized in predicting the main target y_D through a temporal attention mechanism. The whole process is illustrated in Fig. 2.

4.1. Market signal extractor

We employ VAE to recurrently infer and decode the latent factors Z and market movement y from the encoded information X. Although, latent driven factors help in portraying the stock signals leading to the stock movements, the posterior distribution in the generative process given in Eq. (6) usually turns out to be intractable. To estimate these values, variational inference is applied to approximate the posterior $p_\theta(z_d|z_{<d}, x_{\le d}, y_d)$ with a tractable distribution $q_\phi(z_d|z_{<d}, x_{\le d}, y_d)$. The difference between the two distributions is measured by *Kullback–*

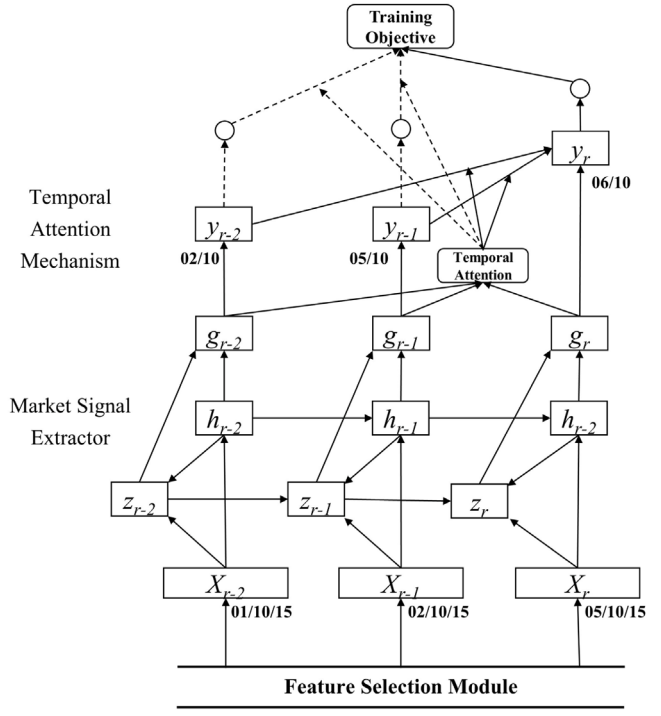

Fig. 2. Illustration of the prediction model. In order to predict stock movement on 06/10/15, we use historical price data from 01/10 ($d-\triangle d$) to 05/10($d-1$). Since 03/10 and 04/10 are not trading days (a weekend), we are left with three days information. We use dashed line to denote auxiliary components. The circles representing temporal objectives are combined with temporal attention mechanism to obtain the final training objective.

Leibler divergence. As given in (Kingma & Welling, 2013; Xu & Cohen, 2018), the recurrent variational lower bound can be given as:

$$\mathcal{L}(\theta,\phi;X,y)=\sum_{d=1}^{D}\mathbb{E}_{q_\phi(z_d|z_{<d},x_{\leq d},y_d)}\{\log p_\theta(y_d|x_{\leq d},z_{\leq d})- D_{KL}[q_\phi(z_d|z_{<d},x_{\leq d},y_d)||p_\theta(z_d|z_{<d},x_{\leq d})]\}\leq\log p_\theta(y|X) \tag{7}$$

where the likelihood term

$$p_\theta(y_d|x_{\leq d},z_{\leq d})=\begin{cases}p_\theta(y_d|x_{\leq d},z_d), & if \quad d<D\\ p_\theta(y_D|X,Z), & if \quad d=D\end{cases}$$

the first term in Eq. (7) denotes the reconstruction likelihood and the second term represents KL divergence. Following the principle of VAE, we try to fit the prior and posterior latent distributions and manage the intractability using *neural approximation and reparametrization* (Kingma & Welling, 2013; Rezende et al., 2014), which aims to maximize the lower bound *w.r.t.* the generative parameter θ and the variational parameter ϕ.

As previously stated stock forecasting is a time series problem, the market signal extractor is implemented as an RNN with GRU (Gated Recurrent Unit) cell to recurrently extract information and decode market signal.

$$h_d^s=GRU(x_d,h_{d-1}^s) \tag{8}$$

Suppose the approximate posterior $q_\phi(z_d|z_{<d},x_{\leq d},y_d)$ to be a multivariate Gaussian distribution with a diagonal covariance structure $\mathcal{N}(\mu,\sigma^2 I)$. The mean and standard deviation are the output of the encoder and can be calculated as:

$$\mu_d=W_{z,\mu}^{\phi}h_d^z+b_\mu^\phi,\qquad \log\sigma_d^2=W_{z,\mu}^{\phi}h_d^z+b_\sigma^\phi \tag{9}$$

and the shared hidden representation can be given as:

$$h_d^z=\tanh(W_z^\phi[z_{d-1},x_d,h_d^{'}y_d]+b_z^\phi) \tag{10}$$

In order to sample from the posterior, the random variable z is expressed as a deterministic variable using *reparametrization* and can be written as:

$$z_d=\mu_d+\sigma_d\odot\epsilon \tag{11}$$

where the $\odot$ represents element-wise product and $\epsilon\sim\mathcal{N}(0,I)$ is the noise term to incorporate stochastic signal in the model. Similarly, we let the prior $p_\theta(z_d|z_{<d},x_{\leq d})\sim\mathcal{N}(\mu,\sigma^2 I)$ and calculate μ_d', σ_d' and $h_d^{z'}$, using independent model parameters and without y_d. Following Zhang et al. (2016) and Xu and Cohen (2018), we set the prior $z_d=\mu_d'$ during decoding and integrate the deterministic features to formulate the final prediction hypothesis as:

$$g_d=\tanh(W_g\cdot[x_d,h_d,z_d]+b_g) \tag{12}$$

$$\tilde{y_d}=\zeta(W_y g_d+b_y)\quad for\quad d<D \tag{13}$$

where W_g and W_y are weight matrices and b_g and b_y are biases. The output of the softmax function ζ measures the confidence over the two classes. The decoding of the main target y_D depends on $z<D$ and is discussed in the next section.

4.2. Temporal attention mechanism

Attention mechanism enables neural networks to focus on relevant segments of feature space to make accurate predictions. Since, each of the temporal auxiliary targets $\tilde{Y}^*=\{\tilde{y}_1;\ldots;\tilde{y}_{D-1}\}$, has varying effect on the training objective and the main target prediction. Therefore, we introduce a shared temporal attention mechanism to discriminate between the effects of different auxiliary outputs and incorporate those effects into the training objective and main prediction hypothesis. Following Xu and Cohen (2018), for the above two scenarios temporal attention computes weights of auxiliary targets by using two separate scoring components: an information score and a dependency score. The information score a_i evaluates lag trading days as per their informational importance, while the dependency score a_d measures the dependency of the main target. The two scoring components can be given as:

$$a_i=w_i^\intercal\tanh(W_{g,i},G^*),\quad a_d=g_D^\intercal\tanh(W_{g,d},G^*) \tag{14}$$

$$a^*=\zeta(a_i\cdot a_d) \tag{15}$$

where $W_{g,i},W_{g,d}\in\mathbb{R}^{d_g\times d_g}$ and $w_i\in\mathbb{R}^{d_g\times 1}$ are model parameters. The integrated representation $G^*=[g_1;\ldots;g_{D-1}]$ and g_R are utilized as the decoded representation of temporal market signal. The two scoring components are combined by feeding their element-wise product into to the softmax function to obtain the normalized attention weight $a^*\in\mathbb{R}^{1\times(D-1)}$. In this way, we accumulate the contribution of temporally-close hypotheses and integrate it into the main hypothesis. The resulting hypothesis for the main prediction target $\tilde{y_D}$ can be formulated as:

$$\tilde{y_D}=\zeta(W_D[\tilde{Y}^*a^{*\intercal},g_D]+b_D) \tag{16}$$

Following Xu and Cohen (2018) and Zhang et al. (2016), we use Monte Carlo method to approximate the expectation term over the posterior in Eq. (7) and use one sample for gradient computation. To incorporate the varying temporal importance at objective level, we first divided the joint training objective $\mathcal{L}(\theta,\phi;X,y)$ into a series of objectives $f\in\mathbb{R}^{D\times 1}$. An objective for a single trading record in the sample observation can be given as:

$$f_d=\log p_\theta(y_d|x_{\leq d},z_{\leq d})-D_{KL}[q_\phi(z_d|z_{<d},x_{\leq d},y_d)||p_\theta(z_d|z_{<d},x_{\leq d})] \tag{17}$$

We reuse the attention vector a^* to build final temporal weight vector $a \in \mathbb{R}^{1\times D}$ and $a = [\alpha a^*, 1]$, where 1 is for the main target and $\alpha = 0.5$ to reduce the effect of auxiliary predictions on model training. Finally, we combine the series of temporal objectives to form our main training objective as:

$$\mathcal{F}(\theta, \phi; X, y) = \frac{1}{N}\sum_{1}^{N} a^{(n)} f^{(n)} \tag{18}$$

Since the objective function in Eq. (18) is differentiable, we take its derivative with respect to model parameters $\{\theta, \phi\}$ through backpropagation for update.

5. Experiments

After computing technical indicators, they are ranked by training L1–LR, SVM and RF using the raw training set. The regression coefficient obtained for each input variable is the associated importance measure of that feature. Each approach is evaluated using *k-fold* cross validation, where the value of k is five. Hyper-parameters λ for $L1$-LR, $\boldsymbol{C}$ for SVM and ***ntree*** for RF are optimized using *GridSearch*. As some of the input features are later dropped in the clustering phase, we have set relatively larger values for $\lambda = 500$, $\boldsymbol{C} = 500$ and $\boldsymbol{ntree} = 1000$.

Since, same variables are ranked differently for different stocks, we calculate the mean variable importance of each variable for the 88 stocks and then scale them between the range [0,1]. Once the feature rankings are computed, it is important to establish a threshold to discard less informative features. In Belanche and González (2011), the authors dropped features having weights two variances farther than the mean importance. On the other hand, Mejía-Lavalle et al. (2006) used the largest gap between consecutive feature weights. Following Belanche and González (2011), we used three variance difference from mean importance and dropped features with variable importance less than 0.03, 0.1 and 0.2 for $L1$-LR, SVM and RF, respectively. The rest of the features are grouped into clusters based on linear correlation among them. Similar to Haq et al. (2019) and Lin et al. (2013), we tested different multiples of median of the similarity matrix as input preferences in the clustering algorithm and selected 11 times the median to generate the desired number of clusters (features). We reuse the variable importance to select the highest ranked feature from each cluster and generate our final feature set.

Input samples for the prediction model are constructed by combining the main observation and any valid trading days in four lag records. The number of hidden nodes is set to 200 for market signal extractor and the biases are initialized to zero. We train the model using Adam optimizer with an initial learning rate of 0.002. Following Xu and Cohen (2018), we use an input drop out rate of 0.3 to regularize the latent variables.

We obtain four different subsets during the feature selection step. The feature sets obtained using the three component methods namely $L1$-LR, SVM, RF, and the final feature set (MFFS) obtained by combining features selected by the three component techniques. We use the original feature set containing the 44 technical indicators and the four subsets selected as input to STOCKNET and compare the results obtained with those of the baseline approaches. As exhaustive comparison is not practical, we select a mixture of traditional and deep learning approaches for comparative analysis. We use a naive predictor making random guess in up and down RAND, an advanced technical analysis approach ARIMA (Brown, 2004), TSLDA (Nguyen & Shirai, 2015). HAN (Hu et al., 2018) and STOCKNET (Xu & Cohen, 2018) as baseline models. In Xu and Cohen (2018), the authors developed four different variations of the model and demonstrated that the HEDGEFUND variant which uses historical price data and tweets data outperforms the others. Therefore, we use the HEDGEFUND variant of the model for comparison and refer to it as STOCKNET.

Following existing studies on stock trend prediction, we use classification accuracy and Mathews Correlation Coefficient (MCC) as evaluation measures. MCC avoids bias due to unbalanced data sets. Given the confusion matrix $\begin{pmatrix} tp & fn \\ fp & tn \end{pmatrix}$, prediction accuracy and MCC can be calculated as:

$$Accuracy = tp + tn/tp + fn + fp + tn \tag{19}$$

$$MCC = tp \times tn - fp \times fn/\sqrt{(tp + fp)(tp + fn)(tn + fp)(tn + fn)} \tag{20}$$

6. Results and discussion

In this section, we first discuss the results of the feature selection approach and then the results of the stock prediction model. In Fig. 3, importance measures for the forty four technical indicators in the raw data set are presented. The figure clearly shows the variations in importance measures as each method employs its own assumption about the underlying relationship between the input and output variables. For example. MACDH, TRANGE and CO ranked the most informative, while CCI, MACDH and LL are considered the least informative features by L1–LR, SVM and RF, respectively.

Features with variable importance measure less than the given threshold are dropped from the three sets of ranked features. [MFI, TYP, MEAN, PCTBB, CCI, DX, WILLR, SK, SD, FK, FD, CO], [MACDH, WILLR, SK, FK, FD] and [MED, TYP, MEAN, HH, LL, MID, MA, EMA, DEMA, WEMA, TEMA, MBB] are dropped from L1–LR, SVM and RF ranked lists, respectively. After clustering the remaining features into groups and selection of delegates from each cluster L1–LR, SVM, and RF selects 14, 9 and 8 features, respectively. Table 2 lists the features selected by each technique and the final set of features generated from them.

Stock trend prediction is a challenging task and minor improvements in prediction accuracy usually lead to large profits. Typically, prediction accuracy of 56% is considered a satisfying result in binary stock prediction (Nguyen & Shirai, 2015). We present the results of different baseline methods and STOCKNET model when combined with different feature selection approaches in Table 3. The mean accuracy and MCC for STOCKNET on different feature sets are produced over 20 independent runs.

Though slightly better than random guess, ARIMA does not achieve satisfactory results with 51.39 accuracy. TSLDA and HAN achieved maximum results of 54.41, 0.653 and 57.64 and 0.051, respectively. STOCKNET is the best performing model among the baseline models with 57.74% mean accuracy and 0.0475 mean MCC. The original data set containing all 44 technical indicators also achieves satisfactory results with 57.04 mean accuracy and 0.0386 mean MCC, when used as an input to STOCKNET. When combined with multi-filter feature selection (MFFS), STOCKNET achieves the best result with 59.44% accuracy and 0.1030 MCC, outperforming STOCKNET by 1.7 in accuracy and 0.055 in MCC. The performance of MFFS confirms the positive effects of combining features selected by multiple disjoint feature selection techniques.

Among the feature selection approaches, RF-based approach selects only 8 input features and has a prediction accuracy 57.43% and MCC 0.0353. SVM-based approach selects 9 features, which were able to achieve prediction accuracy 57.28% and MCC in 0.0249. $L1$-LR-based approach selects 14 features, but has the lowest prediction accuracy among the feature selection approaches. It is important to mention that STOCKNET + MFFS takes only 20 features as input as compared to the large number of input features used by STOCKNET.

A paired t-test is conducted to compare the accuracy scores of the best performing baseline (STOCKNET) and STOCKNET+MFFS. The results show a significant difference in the scores of STOCKNET+MFFS and STOCKNET condition; $t(38) = 7.12$ at 0.05 level of significance. Based on the t-test, we can safely reject the null hypothesis that both are equal. As the results in Table 3 shows that combining deep generative models with feature selection outperforms other approaches having comparatively large number of input features.

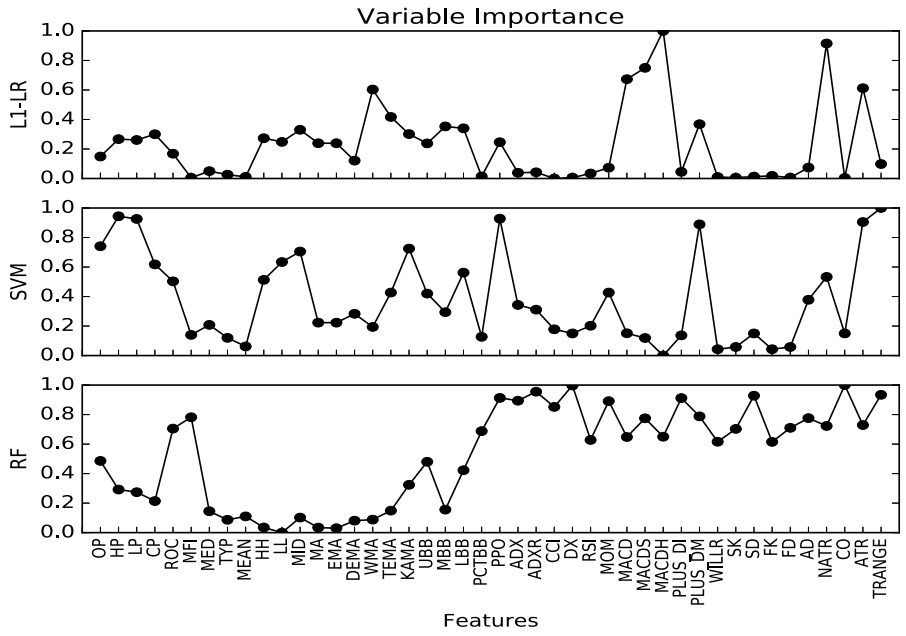

Fig. 3. Variable importance of input features as computed by $L1$-LR, SVM and RF.

Table 2
Selected features.

Approach	d'	Selected Features
L1–LR-based FS	14	**PLUS_DI**, ROC, ADXR, NATR, MACD, MOM, **PPO**, MACDS, MACDH, WMA, **TRANGE**, PLUS_DM, **AD**, RSI
SVM-based FS	9	**TRANGE**, MOM, **AD**, **PPO**, SD, ADX, HP, **PLUS_DI**, CO
RF-based FS	8	**PLUS_DI**, CO, OP, **AD**, DX, **PPO**, SD, **TRANGE**
MFFS	20	MOM, MACDH, WMA, ROC, ADXR, PPO, NATR, TRANGE, MACD MACDS, DX, SK, PLUS_DM, AD, PLUS_DI, RSI, SD, ADX, HP, CO, OP

Note: Features selected by all three approaches are in bold type.
d' is the number of features selected.

Table 3
Results.

Prediction approach	No. of features	Accuracy	MCC
RAND	44	50.89 ± 0.38	−0.0023 ± 0.00
ARIMA	44	51.39 ± 0.00	−0.0206 ± 0.00
TSLDA	Price + Social media	54.41 (max)	0.065382 (max)
HAN	News	57.64 (max)	0.051800 (max)
STOCKNET	Price + Tweets	**57.74** ± 0.36	**0.0475** ± 0.019
STOCKNET + Original Dataset	44	57.04 ± 0.2	0.0386 ± 0.008
STOCKNET + $L1$-LR-FS	14	55.76 ± 0.3	0.0228 ± 0.003
STOCKNET + SVM-FS	9	57.28 ± 0.8	0.0249 ± 0.005
STOCKNET + RF-FS	8	57.43 ± 0.7	0.0353 ± 0.010
STOCKNET + MFFS	19	**59.44** ± 0.96	**0.1030** ± 0.018

6.1. Ablation study

The number of nodes and other hyper-parameters are tuned by adopting the trial and error approach. We tested different number of nodes in the hidden layer and found that optimal results are achieved with 200 nodes. In Table 4, the results of varying number of nodes are presented. The results show that gradual improvement is achieved by increasing the number of nodes up to 200 and any further increase in the number of nodes deteriorates the predictive performance of the learning model.

Table 4
Results with different number of nodes.

Number of nodes	Accuracy	MCC
40	56.94 ± 0.31	0.0141 ± 0.013
100	57.83 ± 0.22	0.0498 ± 0.013
150	58.03 ± 0.37	0.0616 ± 0.015
200	**59.44** ± 0.96	**0.1030** ± 0.018
300	58.89 ± 0.60	0.0621 ± 0.012
400	54.19 ± 0.24	−0.0013 ± 0.031

7. Conclusion

Stock forecasting is an important research topic and has attracted significant attention due to its potential monetary benefits. Accurate prediction of stock price movements strongly depends upon the selection of representative features and developing an appropriate prediction model. Like other machine learning tasks, feature selection is an important part of stock forecasting, but existing studies usually employ a single feature selection technique, which may not consider all assumptions about the underlying regression function linking the

input variables with the output. Therefore, combining features selected by multiple disjoint approaches will select a more optimal feature set and improve the performance of a prediction model. In this study, we demonstrated the effectiveness of an efficient feature selection approach in improving the performance of a deep generative model for stock trend prediction. We combined features selected by $L1$-LR, SVM and RF based feature selection approaches and used the generated feature set as an input to a deep generative model. We tested our approach on a comprehensive data set and demonstrated a carefully selected feature set improves prediction accuracy. Ensemble approach for feature selection enables practitioners to combine the strengths of multiple feature selection techniques and subsequently improve the prediction performance of the learning model. The study also demonstrated the superior performance of deep generated models for stock forecasting as compared to discriminative models.

Considering the weak performance of $L1$-LR based feature selection, performance of the feature selection approach can be further improved by replacing $L1$-LR with another approach. In literature a number of other feature selection techniques can be found, which can be combined to produce a more optimal feature set. In future, we would like to focus on combining other variable ranking approaches to further improve the quality of features selected. Moreover, the impact of different input window length for computing technical indicators on prediction performance also needs further consideration.

CRediT authorship contribution statement

Anwar Ul Haq: Conceptualization, Investigation, Methodology, Writing - original draft. **Adnan Zeb:** Writing - review & editing, Validation. **Zhenfeng Lei:** Resources, Visualization, Validation. **Defu Zhang:** Formal analysis, Writing - review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Adebiyi, A. A., Adewumi, A. O., & Ayo, C. K. (2014). Comparison of arima and artificial neural networks models for stock price prediction. *Journal of Applied Mathematics*, *2014*.

Atsalakis, G. S., & Valavanis, K. P. (2009). Surveying stock market forecasting techniques–part ii: Soft computing methods. *Expert Systems with Applications*, *36*(3), 5932–5941.

Belanche, L. A., & González, F. F. (2011). Review and evaluation of feature selection algorithms in synthetic problems. arXiv preprint arXiv:1101.2320.

Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, *31*(3), 307–327.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.

Brown, R. G. (2004). *Smoothing, forecasting and prediction of discrete time series*. Courier Corporation.

Bustos, O., & Pomares-Quimbaya, A. (2020). Stock market movement forecast: A systematic review. *Expert Systems with Applications*, Article 113464.

Cai, Q., Zhang, D., Wu, B., & Leung, S. C. (2013). A novel stock forecasting model based on fuzzy time series and genetic algorithm. *Procedia Computer Science*, *18*, 1155–1162.

Cervelló-Royo, R., & Guijarro, F. (2020). Forecasting stock market trend: A comparison of machine learning algorithms. *Finance, Markets and Valuation*, *6*(1), 37–49.

Frey, B. J., & Dueck, D. (2007). Clustering by passing messages between data points. *Science*, *315*(5814), 972–976.

Garcia, F., Guijarro, F., Oliver, J., & Tamošiūnienė, R. (2018). Hybrid fuzzy neural network to predict price direction in the german dax-30 index. *Technological and Economic Development of Economy*, *24*(6), 2161–2178.

Ghiassi, M., Saidane, H., & Zimbra, D. (2005). A dynamic artificial neural network model for forecasting time series events. *International Journal of Forecasting*, *21*(2), 341–362.

Gündüz, H., Çataltepe, Z., & Yaslan, Y. (2017). Stock daily return prediction using expanded features and feature selection. *Turkish Journal of Electrical Engineering & Computer Sciences*, *25*(6), 4829–4840.

Guresen, E., Kayakutlu, G., & Daim, T. U. (2011). Using artificial neural network models in stock market index prediction. *Expert Systems with Applications*, *38*(8), 10389–10397.

Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning*, *46*(1–3), 389–422.

Haq, A. U., Zhang, D., Peng, H., & Rahman, S. U. (2019). Combining multiple feature-ranking techniques and clustering of variables for feature selection. *IEEE Access*, *7*, 151482–151492.

Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J., & Scholkopf, B. (1998). Support vector machines. *IEEE Intelligent Systems and their Applications*, *13*(4), 18–28.

Henrique, B. M., Sobreiro, V. A., & Kimura, H. (2019). Literature review: Machine learning techniques applied to financial market prediction. *Expert Systems with Applications*.

Hu, Z., Liu, W., Bian, J., Liu, X., & Liu, T.-Y. (2018). Listening to chaotic whispers: A deep learning framework for news-oriented stock trend prediction. In *Proceedings of the eleventh ACM international conference on web search and data mining* (pp. 261–269).

Huang, C. L., Chen, M. C., & Wang, C. J. (2007). Credit scoring with a data mining approach based on support vector machines. *Expert Systems with Applications*, *33*(4), 847–856.

Kim, K. J. (2003). Financial time series forecasting using support vector machines. *Neurocomputing*, *55*(1–2), 307–319.

Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114.

Lee, M. C. (2009). Using support vector machine with a hybrid feature selection method to the stock trend prediction. *Expert Systems with Applications*, *36*(8), 10896–10904.

Li, B., Wang, Q., & Hu, J. (2011). Feature subset selection: a correlation-based svm filter approach. *IEEJ Transactions on Electrical and Electronic Engineering*, *6*(2), 173–179.

Lin, Y., Guo, H., & Hu, J. (2013). An svm-based approach for stock market trend prediction. In *The 2013 international joint conference on neural networks (IJCNN)* (pp. 1–7). IEEE.

Malkiel, B. G., & Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *The Journal of Finance*, *25*(2), 383–417.

Mejía-Lavalle, M., Sucar, E., & Arroyo, G. (2006). Feature selection with a perceptron neural net. In *Proceedings of the international workshop on feature selection for data mining* (pp. 131–135).

Ng, A. Y. (2004). Feature selection, L 1 vs. L 2 regularization, and rotational invariance. In *Proceedings of the twenty-first international conference on machine learning* (p. 78).

Nguyen, T. H., & Shirai, K. (2015). Topic modeling based sentiment analysis on social media for stock market prediction. In *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: long papers)* (pp. 1354–1364).

Picasso, A., Merello, S., Ma, Y., Oneto, L., & Cambria, E. (2019). Technical analysis and sentiment embeddings for market trend prediction. *Expert Systems with Applications*, *135*, 60–70.

Rakotomamonjy, A. (2003). Variable selection using svm-based criteria. *Journal of Machine Learning Research*, *3*(Mar), 1357–1370.

Rezende, D. J., Mohamed, S., & Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. arXiv preprint arXiv:1401.4082.

Roh, T. H. (2007). Forecasting the volatility of stock price index. *Expert Systems with Applications*, *33*(4), 916–922.

Sezer, O. B., Gudelek, M. U., & Ozbayoglu, A. M. (2020). Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Applied Soft Computing*, *90*, Article 106181.

Trafalis, T. B., & Ince, H. (2000). Support vector machine for regression and applications to financial forecasting. In *Proceedings of the IEEE-INNS-ENNS international joint conference on neural networks. IJCNN 2000. Neural computing: New challenges and perspectives for the new millennium, Vol. 6* (pp. 348–353). IEEE.

Tsai, C. F., & Hsiao, Y. C. (2010). Combining multiple feature selection methods for stock prediction: Union, intersection, and multi-intersection approaches. *Decision Support Systems*, *50*(1), 258–269.

Wu, J. L., Yu, L. C., & Chang, P. C. (2014). An intelligent stock trading system using comprehensive features. *Applied Soft Computing*, *23*, 39–50.

Xu, Y., & Cohen, S. B. (2018). Stock movement prediction from tweets and historical prices. In *Proceedings of the 56th annual meeting of the association for computational linguistics (volume 1: long papers)* (pp. 1970–1979).

Ye, F., Zhang, L., Zhang, D., Fujita, H., & Gong, Z. (2016). A novel forecasting method based on multi-order fuzzy time series and technical analysis. *Information Sciences*, *367*, 41–57.

Zakharov, R., & Dupont, P. (2011). Ensemble logistic regression for feature selection. In *IAPR international conference on pattern recognition in bioinformatics* (pp. 133–144). Springer.

Zhang, D., Jiang, Q., & Li, X. (2007). Application of neural networks in financial data mining. *International Journal of Computer and Information Engineering*, *1*(1), 225–228.

Zhang, B., Xiong, D., Su, J., Duan, H., & Zhang, M. (2016). Variational neural machine translation. arXiv preprint arXiv:1605.07869.

Zhong, X., & Enke, D. (2017). Forecasting daily stock market return using dimensionality reduction. *Expert Systems with Applications*, *67*, 126–139.

Zhong, X., & Enke, D. (2019). Predicting the daily return direction of the stock market using hybrid machine learning algorithms. *Financial Innovation*, *5*(1), 4.