Neural Computing and Applications
https://doi.org/10.1007/s00521-018-3415-3

ORIGINAL ARTICLE

Market impact analysis via deep learned architectures

Xiaodong Li[1] · Jingjing Cao[2] · Zhaoqing Pan[3]

Received: 13 December 2017 / Accepted: 3 March 2018

Abstract
How to deeply process market data sources and build systems to process accurate market impact analysis is an attractive problem. In this paper, we build up a system that exploits deep learning architecture to improve feature representations, and adopt state-of-the-art supervised learning algorithm—extreme learning machine—to predict market impacts. We empirically evaluate the performance of the system by comparing different configurations of representation learning and classification algorithms, and conduct experiments on the intraday tick-by-tick price data and corresponding commercial news archives of stocks in Hong Kong Stock Exchange. From the results, we find that in order to make system achieve good performance, both the representation learning and the classification algorithm play important roles, and comparing with various benchmark configurations of the system, deep learned feature representation together with extreme learning machine can give the highest market impact prediction accuracy.

Keywords Stock prediction · Deep learning · Extreme learning machine

1 Introduction

Financial market, especially the stock market, is quite sensitive to market impacts driven by market events within stock historical prices and market news. How to set up a system to analyze the market impact, in another word, how to analyze the market data and make post-event predictions is an attractive problem to both practitioners and researchers.

Many great efforts have been made in computer science to form the problem within a machine learning framework and provide solutions using classification and regression models. The system built by Schumaker and Chen formulates the problem as a classification task and makes predictions based on textual news articles by analyzing statistical distributions of text features [1]. Yeh et al. [2] use multi-kernel learning model to regress on the stock prices. However, most of these approaches rely on features that are engineered by human experts. Since the complexity within the data can hardly be mined by human labors, it becomes a disadvantage that leads the system to be based on less good representations of the data sets and therefore limits the prediction accuracy of the classification model. How to learn a good feature representation from the raw data automatically and build up a system that can give high prediction performance remains a question to be answered.

Deep learned representation (DLR) which is consisted of a multiple-layer architecture has the ability to use multiple levels of abstraction in a unsupervised way to learn a feature representation that is less sensitive to data invariant [3]. Different from human-engineered features or supervised/unsupervised feature selection approaches, at each layer of DLR, it adopts nonlinear but simple activation functions as nodes to transform the input feature representation into a representation that is more abstract. Thus, the feature representation of higher level can provide a solid base for future classification tasks. This advantage of DLR can be applied to financial data, i.e., market news and

✉ Jingjing Cao
bettycao@whut.edu.cn

Xiaodong Li
xiaodong.c.li@outlook.com

[1] College of Computer and Information, Hohai University, Nanjing, China

[2] School of Logistics Engineering, Wuhan University of Technology, Wuhan, China

[3] School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China

Published online: 13 March 2018

prices, to construct more abstract features than traditional ones.

For the classification model, extreme learning machine (ELM) has been proved to have high prediction accuracy in tasks that deal with many data analysis problems [4, 5]. Unlike neural networks, as shown by Huang et al., the internal weights and hidden layer biases are randomly assigned by ELM instead of a fully tuning process of all the internal parameters [5]. Specifically, ELM could analytically determine the output weights with the configuration of single-hidden-layer feedforward networks (SLFNs), which has also been proved to give the smallest norm of weights (Theorems 2.1 and 2.2 in [5]). The kernelized version of ELM is proposed by Huang and Siew [4], which improves ELM's prediction accuracy to a higher level and has been examined by many real-world data sets [6, 7].

Considering the good properties of deep learning in feature representation and ELM in classification, we build up a market impact analysis system which employs DLR to do feature abstraction based on raw or shallowly preprocessed market data in order to improve the feature representation, and adopts ELM to integrate deeply learned features to make more accurate predictions than conventional ones. Different system configurations, i.e., different feature learning modules, classification modules and different settings within each module, are explored and tested by 1-year intraday tick prices and news articles in Hong Kong Stock Exchange. Experimental results demonstrate the effectiveness of the proposed system.

The rest of this paper is organized as follows. Section 2 reviews the usage of deep learning in feature representation, extreme learning machine and its applications. Section 3 presents our proposed system. Section 4 reports the experimental comparisons and discussions. Section 5 gives the conclusion and future work directions.

2 Related work

In the first part of this section, we review previous works that are related to deep learned representation and its applications. In the second part, we summarize the works that are related to extreme learning machine. In the third part, we review the approaches and systems that are used to do market impact analysis.

2.1 Deep learned representations

There are many solid research results and successful applications that use deep learning algorithms to improve feature representations [3, 8–10], especially in image processing and speech recognition [11–16]. Huang and LeCun [17] present a hybrid system where invariant image features are learned by a trained convolutional network, and object classification task is then conducted by a Gaussian kernel SVM based on learned features. In [18], Lee et al. present the convolutional deep belief network using probabilistic max-pooling technique which can shrink higher layer representations so that the model can scale to realistic image size. Then, in [19], Lee et al. apply convolutional deep belief networks to audio data and use various audio classification tasks to empirically evaluate their model.

Besides the success of deep learning models in image processing and speech recognition, there are also many works related to text mining. In [20], recursive auto-encoders have been proposed to beat the state-of-the-art full sentence paraphrase detection algorithms. The model generalizes recurrent networks and almost doubles the $F-1$ score of paraphrase detection task. In [21, 22], sentiment features are also successfully improved by deep learned representations. In [23], textual representation learning is used to solve word sense disambiguation, where on the subset of Senseval-3 the model improves the accuracy from 67.8 to 70.2%.

Deep learning models in finance applications have also attracted many interests. In [24], Heaton et al. explore the use of auto-encoder in portfolio construction. A four-step algorithm for building portfolios is introduced, where deep learning is employed as a unsupervised feature extractor and another classifier is stacked on the deep learning model to decide which stock has a larger chance to outperform the market performance. In [25], Krauss et al. compare the performance of deep neural networks and tree-based models in the context of statistical arbitrage trading. They find that equal weight of the selected models gives the best performance. In [26], Chong et al. apply deep learning networks to stock market analysis and prediction. The feature extraction performances of principle component analysis and auto-encoder on the high-frequency trading tick raw data are empirically compared, which is similar to our approach in this paper except that we exploit two heterogeneous data sources i.e., news articles and tick data.

2.2 Extreme learning machine

Extreme learning machine (ELM) is proposed by Huang et al. [5], which has been successfully used as a supervised learning model in many tasks [27]. Sun et al. [28] apply ELM to fashion retail sales predictions and find the performance better than BP-NN. Sun et al. [29] apply OS-ELM to P2P networks based on an ensemble classification framework. In the field of bioinformatics, Handoko et al. [30] use ELM to predict peptides binding to the human leukocyte antigens (HLA). Sarawathi et al. [31] use a combination of integer-coded genetic algorithm (ICGA)

and particle swarm optimization (PSO) together with ELM for gene selection and classification. Huang et al. [32] employ ELM and apply it to traffic sign detection task. There are many works combining deep learning and ELM to achieve better performance in their data sets. Using auto-encoder and ELM, Kasan et al. [33] give results that outperform many other state-of-the-art deep learning methods in MNIST OCR data set. In [34], Tang et al. use deep neural networks and ELM in ship detection on spaceborne images, where deep neural networks are used for higher-level feature representation and ELM is used for decision-making. In [35], Li et al. applied ELM to both market news and prices to make predictions of price movements.

2.3 Market impact analysis

Market news and stock prices are two of the most important sources of market information that are used for market impact analysis. In [36], Seo et al. follow the approaches of text mining and build a multi-agent system for intelligent portfolio management which assesses companies' risk levels by analyzing textual news features. In [1, 37], Schumaker and Chen propose the AZFinText system, which is based on terms' statistical model of news, to make predictions of future market price movements. In [38], multi-document summarization algorithm is first applied to news and more accurate predictions are generated based on the summaries instead of the original full text. Besides the works on news, there have been great efforts to analyze the market impact based on market prices. In [39], Gestel et al. apply support vector regression to price and make predictions on volatility. In [40–44], Tay and Cao improve their previous works where the objective function of support vector machine is modified to adapt non-stationary price time series. In [45], Huang et al. propose a support vector machine-based system to predict the price movements of NIKKEI 225 index.

In [46], a market making trading strategy is proposed, which places orders in the market order book based on the signals generated from market quote ticks. In [47], Li et al. enhance the market impact prediction accuracy by integrating news and price information sources.

In this paper, we take the advantages of the deep learned representations as reviewed in Sect. 2.1 and apply it to market news and stock tick prices in order to have a better feature representation than the human-engineered one. On the other hand, we consider the good classification performance of extreme learning machine as stated in Sect. 2.2, and set up a market impact analysis system which has the extreme learning machine on top of the deep learned representations. We design several different system configurations to compare the performances of the proposed system with many benchmarks. The empirical results indicate that the proposed system produces convincing outputs and outperforms the benchmarks in most of the experimental cases.

3 Deep learned architecture

The architecture of the market impact analysis system is shown in Fig. 1. In this section, we explain the processing pipeline of the system step by step. The whole system consists of two parts: (1) unsupervised deep learned representation and (2) supervised classification. The first part uses multiple layers of auto-encoders to do abstraction on the input data, and the second part utilizes several different machine learning models to make market impact predictions based on the features generated in the first part.

3.1 Preprocessing of prices and news

Some of the news articles are to be filtered out because of two constraints: (1) market trading hours and (2) prediction horizon overlaps. Take Hong Kong Stock Exchange for example, the trading hour starts from 9:30 to 12:00 in the morning and 1:00–4:00 in the afternoon. Since news impact overnight is believed to be absorbed in the morning auction hour, the system in this paper only keeps the news articles that have time stamps within the trading hours. The second issue is the overlap of prediction horizons. As illustrated in Fig. 2, assume two news articles d_1 and d_2 on the same company are released within time window Δ, it is hard to determine whether the market impact at time $t_{+\Delta}$ is generated by d_1, or d_2, or both. To avoid this situation, d_1 is purposely eliminated in the preprocessing. Following the approach of news preprocessing in text mining [48], news texts are firstly preprocessed Chinese segmentation and stop word filtering. In the second step, each word is considered as a shallow feature, and TF.IDF (term frequency and inverse document frequency) [49] is calculated as the weight of features.

To the best of our knowledge, it is hard to theoretically determine a prediction horizon for each news piece. In order to cover more cases, post-news 5, 10, 15, 20, 25 and 30 min are used to mimic different prediction horizons. The snapshot prices at the time points are extracted and converted into *simple returns* which are further discretized and used as labels of the news articles. To formulate, assume one piece of news is at t_0, and the snapshot price at t_0 is p_0. According to the determination method of the prediction horizon, future 5-, 10-, 15-, 20-, 25- and 30-min prices are extracted, denoted as p_{+5}, p_{+10}, p_{+15}, p_{+20}, p_{+25}, and p_{+30}, respectively, and *simple return* is calculated by Eq. (1),

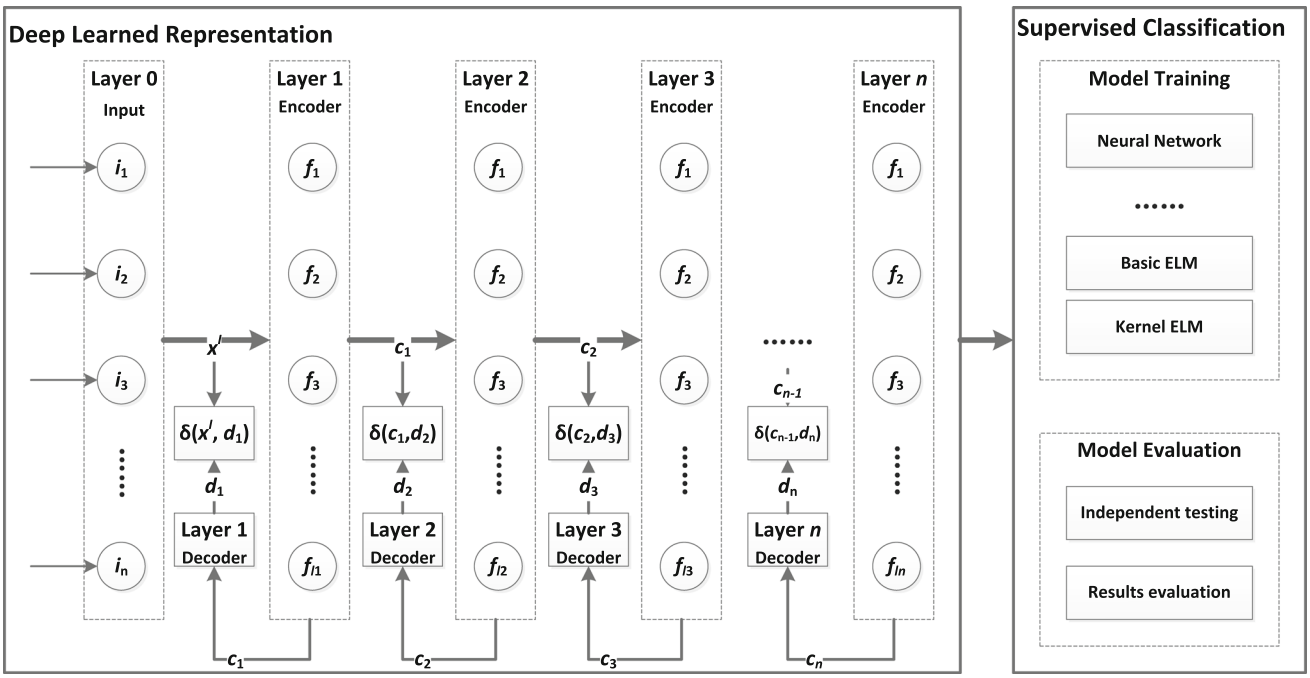

Fig. 1 Deep learned system architecture

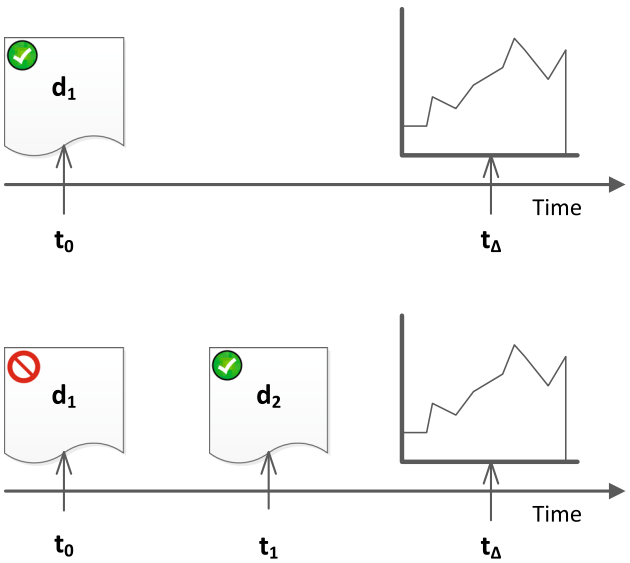

Fig. 2 News filtering

$$r = \frac{p_i - p_0}{p_0}. \tag{1}$$

r is further discretized by thresholds $\pm\theta$, and the detailed label construction method is discussed in Sect. 4.1.

It has been empirically discovered in [50] that prediction accuracy can be improved by combining both market news and price features. Therefore, in this paper, we set up three experiment groups using

$$\phi : \{\mathbf{Text}\} \mapsto \mathbb{L}, \tag{2}$$

$$\phi : \{\mathbf{Price}\} \mapsto \mathbb{L}, \tag{3}$$

$$\phi : \{\mathbf{Text}, \mathbf{Price}\} \mapsto \mathbb{L}, , \tag{4}$$

respectively, where **Text** denotes the textual features of news, **Price** denotes the features of prices, and $\mathbb{L}$ is the set of prediction labels.

3.2 System formulation

Descriptions of mathematical symbols used in this formulation are listed in Table 1.

In the DLR part, input instances at layer 0 of the auto-encoder is $x^I \in R^d$, where I stands for *Input* and d the dimension of features. Within layer 0, each node i_i is an identity function which does not change the input instances. The output of layer 0 is taken as the input of layer 1. Function f_i in each node of layer 1 can be chosen from many candidates which are simple but not linear functions, such as a sigmoid function. Using E_1 to denote the functions in layer 1 as a whole, it firstly transits the input x^I into code c_1,

$$E_1(x^I) = c_1. \tag{5}$$

Secondly, c_1 (together with a constant bias) is used by a decoder D to reconstruct the input instances,

Table 1 Mathematical symbols and their meanings

Symbol	Meaning
Deep learned representation part	
x^I	Input instances
E	Encoder function
c	Code after encoder
D	Decoder function
d	Decode after decoder
δ	Error function
Supervised classification part	
t	Target labels
g	Activation function
$\mathbf{w}$	Input weight
b	Bias of ith hidden neuron
β	Output weight
$\mathbf{H}$	Hidden layer output matrix

$$D(c_1) = d_1. \tag{6}$$

d_1 is not the same as x^I, and the object of auto-encoder is to minimize the *distance* between d_1 and x^I, such as to minimize the square error in Eq. 7 and back-propagate the error to the encoder in order to tune the weights of the layer,

$$\delta(d_1, x^I) = \|d_1 - x^I\|^2. \tag{7}$$

After training, f_i and related weights are tuned, and the final outputs of layer 1 are the best summary of the input instances. In this way, the outputs of $k-1$ layer are compressed and abstracted by layer k, and n layers of the auto-encoders are stacked to construct a DLR.

In the supervised classification part, suppose the outputs of DLR have N arbitrary distinct instances $(\mathbf{x}_i, \mathbf{t}_i)$, where $\mathbf{x}_i \in \mathbf{R}^{l_n}$ and $\mathbf{t}_i \in \mathbf{R}^m$. The standard SLFNs with $\tilde{N}$ hidden neurons and activation function $g(x)$ are mathematically modeled as

$$\sum_{i=1}^{\tilde{N}} \beta_i g(\mathbf{w}_i \cdot \mathbf{x}_j + b_i) = \mathbf{o}_j,\ j = 1, \ldots, N, \tag{8}$$

where $\mathbf{w}_i$ is the input weight, b_i is the bias of ith hidden neuron, and β_i is the output weight. When zero-error approximation, i.e.,

$$\sum_{j=1}^{\tilde{N}} \|\mathbf{o}_j - \mathbf{t}_j\| = 0, \tag{9}$$

Equation (8) could be reframed as

$$\mathbf{H}\beta = \mathbf{T}, \tag{10}$$

where $\mathbf{H}$ is the hidden layer output matrix of the neural network.

One of the advantages of ELM is that it has universal approximation capability [51], which denotes that ELM can approximate any continuous target functions. Given any target continuous function T, there exists a series of β_i such that

$$\lim_{j \to +\infty} \|\mathbf{o}_j - \mathbf{t}_j\| = \lim_{j \to +\infty} \left\| \sum_{i=1}^{\tilde{N}} \beta_i g(\mathbf{w}_i \cdot \mathbf{x}_j + b_i) - \mathbf{t}_j \right\| = 0 \tag{11}$$

which is the same as to minimize the error

$$\min_{\mathbf{w}, b, \beta} \sum_{j=1}^{N} \left(\sum_{i=1}^{\tilde{N}} \beta_i g(\mathbf{w}_i \cdot \mathbf{x}_j + b_i) - \mathbf{t}_j \right)^2. \tag{12}$$

It can be proved that classification capability of the generalized SLFNs with the hidden-layer mapping $\mathbf{H}$, which is similar with the classification capability theorem of single-hidden-layer feedforward neural networks, satisfying the universal approximation condition.

Different from gradient-based learning algorithms, we use Huang's solution that does not need to tune all the parameters within the network [5], which is to solve a linear system to give the smallest norm least-square solution, i.e.,

$$\hat{\beta} = \mathbf{H}^{\dagger}\mathbf{T}. \tag{13}$$

In this paper, a RBF (radial basis function) kernel-based ELM is also adopted in the system configuration. Equation (10) could be formulated as

$$\sum_{i=1}^{\tilde{N}} \beta_i \exp\left(\frac{\|\mathbf{x}_j - \mu_i\|^2}{\sigma_i} \right) = \mathbf{t}_j,\ j = 1, \ldots, N, \tag{14}$$

where Gaussian kernel is included in ELM, and it has the ability of infinite differential. $\hat{\beta}$ can be solved by Eq. (13).

4 Experiments and discussions

4.1 Data sets

Two financial data sets are used in the experiment, i.e., market news and stock tick prices used in [47]. The 1-year market news archive is bought from Caihua Agency,[1] where each news piece is tagged by a time stamp indicating its release time. The stock tick prices are bought from Hong Kong Stock Exchange,[2] where all the company

[1] www.finet.hk.

[2] www.hkex.com.hk.

names listed in Hong Kong market of year 2001 are included.

In the experiment, we select a proportion of all the stocks that are liquid stocks listed in Hang Seng Index constituents (HSI).[3] According to HSI update log, HSI has 33 stocks in year 2001, and two updates of the HSI happened on 1 June and 31 July. Due to the *tyranny of indexing* [52], we select the stocks that have been HSI constituents throughout 2001. Therefore, the total number of stocks selected is 23.

There are 6 prediction horizons from 5 min (5 m) after the news is released to 30 min (30 m), and the numbers of instances for each prediction horizon are listed in Table 2.

Each instance is labeled by comparing the short-term price return with a predefined threshold θ, indicating whether the prices go up (+ 1) after the news, remain in a small range (0), or decrease (− 1),

$$l(x) = \begin{cases} +1 & \text{if } \mathrm{r}(\mathrm{x}) \geq \theta \\ -1 & \text{if } \mathrm{r}(\mathrm{x}) \leq -\theta \\ 0 & \text{otherwise} \end{cases}, \tag{15}$$

where $l(x)$ is the label of instance x, and $r(x)$ is the simple return of x.

We evaluate models' performance by measuring their prediction accuracy, which is formulated as follows:

$$\mathrm{acc} = \frac{\mathrm{hits}}{\mathrm{all}}, \tag{16}$$

and

$$\mathrm{hits} = t_{++} + t_{00} + t_{--}, \tag{17}$$

$$\mathrm{all} = t_{++} + t_{00} + t_{--} + f_{0+} + f_{-+} + f_{+0} + f_{-0} + f_{+-} + f_{0-}, \tag{18}$$

where t_{++}, t_{00}, t_{--}, f_{0+}, f_{-+}, f_{+0}, f_{-0}, f_{+-} and f_{0-} are defined in Table 3.

Both the value of θ and the method that discretizes the price returns largely influence the distributions of the instances among classes. Suppose we arbitrarily choose a θ and follow the hypothesis in finance domain that the distribution of stock price returns is Gaussian with *fat tail*, the true label −1, 0 and +1 have following probabilities,

$$P_{-1} = \int_{-\infty}^{-\theta} \mathrm{pdf}_{\mathrm{Gaussian}}(x)\, \mathrm{d}x, \tag{19}$$

$$P_0 = \int_{-\theta}^{\theta} \mathrm{pdf}_{\mathrm{Gaussian}}(x)\, \mathrm{d}x, \tag{20}$$

$$P_{+1} = \int_{\theta}^{+\infty} \mathrm{pdf}_{\mathrm{Gaussian}}(x)\, \mathrm{d}x. \tag{21}$$

Given the label distribution without extra learning, people can conduct a random draw based on the distribution and make predictions. If the random draw prediction is eventually the same as the true label, hits in acc will increase. To formulate, acc can be calculated by

$$\mathrm{acc} = P_{-1}^2 + P_0^2 + P_{+1}^2. \tag{22}$$

Since θ and $-\theta$ are symmetric around 0,

$$P_{-1} = P_{+1}, \tag{23}$$

$$P_0 = 1 - 2P_{+1}, \tag{24}$$

substitute Eqs. (23) and (24) into (22), and denote P_{+1} as P, Eq. (22) becomes

$$\mathrm{acc} = 6P^2 - 4\,\mathrm{P}\, + 1. \tag{25}$$

If we plot acc along with the change in θ in Fig. 3,

Table 2 Numbers of instances for each prediction horizon

Prediction horizon	5 m	10 m	15 m	20 m	25 m	30 m
Total	1721	1953	2035	1965	1963	1906

Table 3 Definition of t_{++}, t_{00}, t_{--}, f_{0+}, f_{-+}, f_{+0}, f_{-0}, f_{+-} and f_{0-}

	Predict +	Predict 0	Predict −
True +	t_{++}	f_{+0}	f_{+-}
True 0	f_{0+}	t_{00}	f_{0-}
True −	f_{-+}	f_{-0}	t_{--}

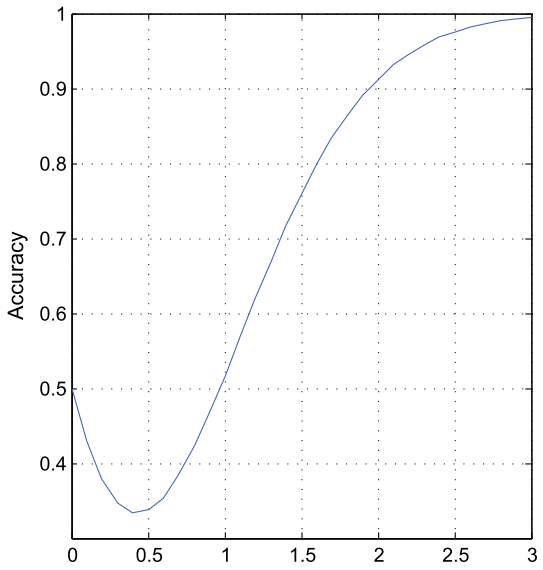

Fig. 3 Accuracy along with θ

[3] www.hsi.com.hk.

Table 4 Independent testing accuracy results

Ftr.	Cls.	5m		10m		15m		20m		25m		30m	
		Avg.	Dev.	Avg.	Dev.	Avg.	Dev.	Avg.	Dev.	Avg.	Dev.	Avg.	Dev.
Text													
PCA	BELM	0.5201	0.0362	0.5019	0.0481	0.5026	0.0407	0.4994	0.0512	0.4916	0.0279	0.5120	0.0283
PCA	KELM	0.5114	0.0392	0.5158	0.0335	0.5034	0.0420	0.4882	0.0478	0.5054	0.0440	0.4935	0.0361
DLR	NN	0.4987	0.0257	0.5127	0.0456	0.4866	0.0293	0.4882	0.0442	0.4970	0.0421	0.5116	0.0400
DLR	BELM	0.5040	0.0326	0.5139	0.0369	0.5233	0.0351	0.5065	0.0394	0.5144	0.0481	0.5148	0.0315
DLR	KELM	**0.5349**	0.0410	**0.5791**	0.0282	**0.5845**	0.0388	**0.5609**	0.0481	**0.5749**	0.0297	**0.5343**	0.0510
Price													
PCA	BELM	0.5020	0.0377	0.4880	0.0453	0.5220	0.0300	0.4982	0.0373	0.5018	0.0302	0.5028	0.0214
PCA	KELM	0.4946	0.0327	0.4937	0.0369	0.5293	0.0455	0.5041	0.0439	0.5042	0.0495	0.5079	0.0327
DLR	NN	0.5161	0.0376	0.4867	0.0556	0.4862	0.0305	0.4787	0.0322	0.5042	0.0362	0.5176	0.0289
DLR	BELM	**0.5510**	0.0488	0.5551	0.0390	0.5349	0.0265	0.5296	0.0400	0.5204	0.0399	0.5324	0.0196
DLR	KELM	0.5443	0.0361	**0.5576**	0.0442	**0.5483**	0.0275	**0.5432**	0.0494	**0.5359**	0.0473	**0.5366**	0.0365
Text+Price													
PCA	BELM	0.4946	0.0422	0.4981	0.0444	0.5073	0.0293	0.4917	0.0562	0.4946	0.0332	0.4991	0.0352
PCA	KELM	0.5054	0.0523	0.4861	0.0437	0.4996	0.0325	0.4781	0.0248	0.4766	0.0388	0.5088	0.0446
DLR	NN	0.2651	0.0493	0.2443	0.0816	0.2349	0.0669	0.2136	0.0499	0.2575	0.0814	0.2347	0.0673
DLR	BELM	0.5134	0.0401	0.5152	0.0334	0.5207	0.0221	0.5095	0.0478	0.5108	0.0427	0.5046	0.0392
DLR	KELM	**0.5463**	0.0405	**0.5892**	0.0259	**0.5754**	0.0342	**0.5793**	0.0408	**0.5581**	0.0343	**0.5523**	0.0239

Bold values indicate the best performers in the test group. Underlined values indicate the second best performers in the test group

we find that the chart has a upside-down sine function shape, with the minimum value at $P = \frac{1}{3}$, which is equivalent to flipping a fair coin with three outcomes for the prediction. On the other side, acc increases along with θ on the right-hand side of the minimum point, which means when the value of θ is great, most of the instances are labeled with 0 and the probability of correct guess becomes high. In another word, the accuracy can be increased only by manipulating the labeling method without any change in the learning model. Based on the explanations, θ cannot be too large as it will bring too much acc increment. On the other hand, θ cannot be too small, since θ should be greater than the basic market transaction cost which is 0.003 (30 bps) in Hong Kong market. To balance both the constraints, we choose $\theta = 0.003$ (30 bps).

4.2 Model setup and parameter tuning

The experiment is separated into three groups, which uses Text data set, Price data set and Text+Price data set, respectively. In the Text data set, there are 1000 textual shallow features derived from word frequency statistics, while in the Price data set, there are 11 preliminary technical indicator features based on market tick prices. In the Text+Price data set, we simply combine the features of Text and Price and form a feature vector of 1011 dimensions. Since instances in the Price data set are sampled at the time point of news release, the number of the instances is the same in both the Text and Price data sets.

Models and benchmarks that are examined in each group are set up along two dimensions. The first dimension is feature learning model (ftr.), where it includes (1) DLR and (2) principle component analysis (PCA). The second dimension is supervised classification model (cls.), where it includes (1) basic extreme learning machine (BELM), (2) kernel extreme learning machine (KELM) and (3) neural networks (NN). The detailed model setup is illustrated as follows:

- *PCA+BELM, PCA+KELM* We use PCA as a benchmark feature selection algorithm in the experiment. PCA is trained and 85% of cumulative energy is reserved. The output features of PCA are further fed into BELM for classification. BELM has one parameter to be tuned, which is the number of hidden nodes. We follow the guideline in [35] and set the number between the number of output features of PCA and the number of the instances. The RBF kernel is adopted for KELM. We use grid search technique for kernel's parameters, where γ searches in the range $\{2^{-17}, 2^{-16}, \ldots, 2^{2}\}$ and C searches in the range $\{2^{-5}, 2^{-4}, \ldots, 2^{14}\}$.
- *DLR+NN, DLR+BELM, DLR+KELM* The DLR has 10 layers in the architecture, and each layer is an auto-encoder. For comparison, the number of the outputs in

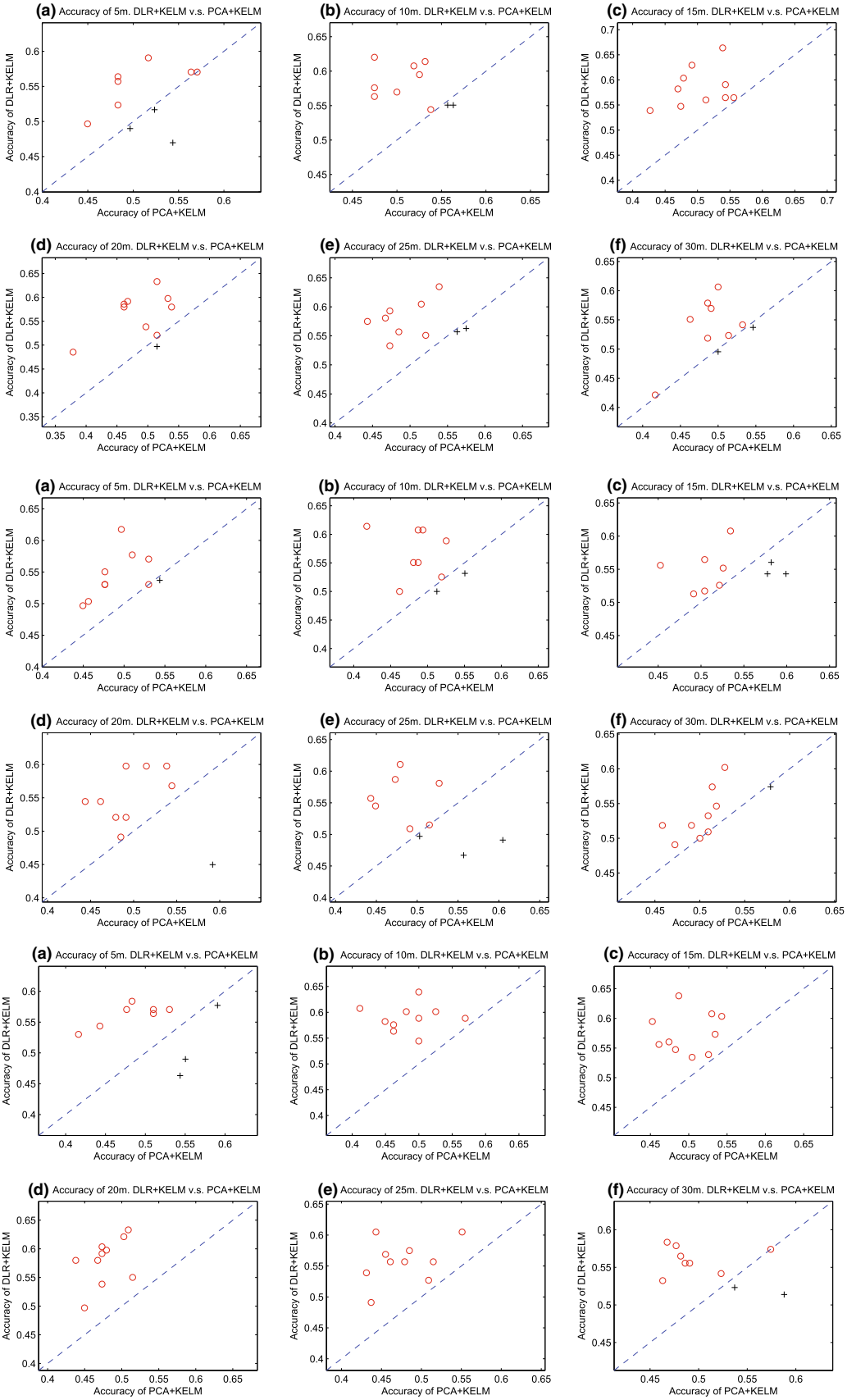

Fig. 4 One-to-one comparison between DLR+KELM and PCA+KELM: 1 Text, 2 Price and 3 Text+Price

Table 5 Accuracy improvement at different prediction horizons, using DLR+KELM

	5 m	10 m	15 m	20 m	25 m	30 m
Text+Price v.s. Text						
Text	0.5349	0.5791	0.5845	0.5609	0.5749	0.5343
Text+Price	0.5463	0.5892	0.5754	0.5793	0.5581	0.5523
Improve (%)	2.1	1.7	− 1.5	3.3	− 2.9	3.4
Text+Price v.s. Price						
Price	0.5443	0.5576	0.5483	0.5432	0.5359	0.5366
Text+Price	0.5463	0.5892	0.5754	0.5793	0.5581	0.5523
Improve (%)	0.4	5.7	5.0	6.6	4.1	2.9

The first two rows in each sub-table are the accuracy results, and the last row in each sub-table is the accuracy improvement

the 10th layer is set the same as the output feature number of PCA. In each round of experiment, the number of output features in the PCA changes due to different data sets. Following the PCA, DLR changes the number of the output features accordingly, based on which, the number of hidden nodes in DLR linearly decreases from the 1st layer until the 10th layer. The implementation of NN in deep learning toolbox[4] is adopted in this setting. The setup of BELM in DLR+BELM and KELM in DLR+KELM is the same as the setup of BELM in PCA+BELM and KELM in PCA+KELM, respectively.

For each prediction horizon, 80% of the instances are randomly selected as the training data set, 10% the validation data set, and 10% the independent testing data set. The training–validation–testing process runs 10 rounds for each model in each prediction horizon.

4.3 Experimental results and findings

The average accuracy and standard deviation of independent testing results are listed in Table 4. The best performer in each group of each prediction horizon is marked in bold font, and the second best performer is underlined. Firstly, it could be observed that among 18 (6 prediction horizons × 3 groups) columns, DLR+KELM achieves 17 best performers and 1 second best performer, and DLR+BELM achieves 14 second best performers, which means systems using deep learned features together with extreme learning machine models outperform the other system configurations in the experiment setting.

Secondly, to compare the effectiveness of feature learning part, we fix the classification model part (using KELM) and use 10 rounds results to do one-to-one comparisons between DLR+KELM and PCA+KELM. The results are plotted with a scatter graph in Fig. 4. The figure has three groups of subgraphs separated by horizontal lines, corresponding to three different data sets. Red circle means that DLR+KELM outperforms PCA+KELM in the prediction horizon, and black cross means the opposite. From the results, it can be observed that in most of the cases, DLR+KELM has better prediction accuracy.

Thirdly, with the same deep learned features, however, DLR+NN does not give good results. It achieves no best performer or the second best. Based on this observation, it can be concluded that to achieve a market impact analysis system with good performance, both the feature learning and the classification model play critical roles.

Besides the comparisons between different models, we select the results of the best performer DLR+KELM and compare the accuracies between different data sources. As illustrated in Fig. 5, results of DLR+KELM over different prediction horizons based on different data sources are represented by different lines. It can be observed that except two points, models with features from Text+Price data sources outperform the models that employ only one data source. To quantify the differences, we list the accuracy results of Text, Price and Text+Price using DLR+KELM in Table 5 and calculate the improvements in Text+Price v.s. Text and Text+Price v.s. Price, respectively, by,

$$\text{Improve}(x) = \frac{\text{TP} - x}{x} \times 100\%, \tag{26}$$

where TP is the accuracy of "Text+Price," and x can be the accuracy of either Text or Price.

From the results, we can see that: (1) Comparing with the improvements in Text+Price from Text, the improvements from Price are greater. Two under-perform points in Text+Price v.s. Text are at 15 and 25 m, which means adding Price data source at longer prediction horizon has negative effects to the system, which indicates that the prediction power of Price starts to vanish from 15 m; (2) The improvement in Text+Price v.s. Price at 5m is minor, which means including Text at 5 m gives few positive

[4] http://cn.mathworks.com/matlabcentral/fileexchange/38310-deep-learning-toolbox.

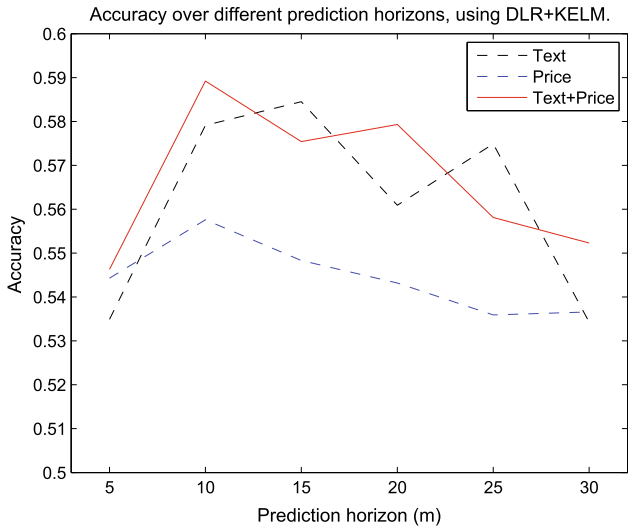

Fig. 5 Accuracy over different prediction horizons, using DLR+KELM

effect in short prediction horizon, in another word, the prediction power of Text does not take effect in short-term prediction.

5 Conclusion and future work

In this paper, we build up a system that exploits deep learning architecture and adopt state-of-the-art supervised learning algorithm—extreme learning machine—to predict market impacts. Several configurations of representation extraction/abstraction and supervised learning algorithms, i.e., PCA+BELM, PCA+KELM, DLR+NN, DLR+BELM and DLR+KELM, are evaluated in the experiments which are conducted on the intraday tick-by-tick data of 23 stocks in the H-share market and corresponding commercial news archives. It has been shown that using the configuration of deep learned representation and extreme learning algorithm can possibly help improve the prediction accuracy of the system. From the empirical results, we find that

- Overall, DLR+KELM largely outperforms the other four system configurations, and DLR+BELM achieves the second best performer.
- While fixing the classification model part, configuration with DLR has better prediction accuracy than that with PCA, which means DLR makes better feature abstraction in the task.
- While fixing the feature representation part, configuration with NN does not give any better result, which means to make system have good performance, the classification model part also plays a critical role.
- Observing along the prediction horizon, the predictability based on news tends to have more power at future 15 min, and that based on prices tends to reach its maximum in future 5–10 min.

In the system, the combination of deep learned feature representations and classification models is designed in a loosely coupled way, where the information loss between true and predicted labels on the classification part cannot be properly back-propagated to the deep learned architecture part, which reduces the learning ability of the network as a whole. In the future work, how to integrate those two parts in an intelligent way, and let the deep learning representation make more use of the label information is worth investigation.

Acknowledgements The work described in this paper was partially supported by National Natural Science Foundation of China under the Grant Nos. 61602149 and 61502360, partially supported by the Fundamental Research Funds for the Central Universities under the Grant No. 2016B01714, and partially supported by Priority Academic Program Development of Jiangsu Higher Education Institutions.

Compliance with ethical standards

Conflict of interest We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work; there is no professional or other personal interest of any nature or kind in any product, service and/or company that could be construed as influencing the position presented in, or the review of, the manuscript entitled, "Market Impact Analysis via Deep Learned Architectures."

References

1. Schumaker RP, Chen H (2010) A discrete stock price prediction engine based on financial news. Computer 43(1):51–56
2. Yeh C-Y, Huang C-W, Lee S-J (2011) A multiple-kernel support vector regression approach for stock market price forecasting. Expert Syst Appl 38:2177–2186
3. Lecun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521:436–444
4. Huang G-B, Siew C-K Extreme learning machine: RBF network case. In: Control, automation, robotics and vision conference, ICARCV'04
5. Huang G-B, Zhu Q-Y, Siew C-K (2006) Extreme learning machine: theory and applications. Neurocomputing 70(1):489–501
6. Wei X-K, Li Y-H, Feng Y Comparative study of extreme learning machine and support vector machine. In: Advances in neural networks, ISNN'06
7. Huang G-B, Wang DH, Lan Y (2011) Extreme learning machines: a survey. Int J Mach Learn Cybern 2(2):107–122
8. Bengio Y (2009) Learning deep architectures for AI. Found Trends Mach Learn 2(1):1–127
9. Schmidhuber J (2015) Deep learning in neural networks: an overview. Neural Netw 61:85–117
10. Zhang H, Chow TWS, Wu QMJ (2016) Organizing books and authors by multilayer som. IEEE Trans Neural Netw Learn Syst 27(12):2537–2550

11. Chen Y-N, Han C-C, Wang C-T, Jeng B-S, Fan K-C (2006) The application of a convolution neural network on face and license plate detection. In: International conference on pattern recognition, vol 3, pp 552–555
12. Ranzato M, Huang FJ, Boureau Y-L, LeCun Y (2007) Unsupervised learning of invariant feature hierarchies with applications to object recognition. In: 2014 IEEE conference on computer vision and pattern recognition, pp 1–8
13. Larochelle H, Erhan D, Courville A, Bergstra J, Bengio Y (2007) An empirical evaluation of deep architectures on problems with many factors of variation. In: Proceedings of the 24th international conference on machine learning, ICML'07, New York, NY, USA. ACM, pp 473–480
14. Vincent P, Larochelle H, Bengio Y, Manzagol P-A (2008) Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th international conference on machine learning, ICML'08, New York, NY, USA. ACM, pp 1096–1103
15. Zhang H, Cao X, Ho JKL, Chow TWS (2017) Object-level video advertising: an optimization framework. IEEE Trans Ind Inform 13(2):520–531
16. Zhang H, Li J, Ji Y, Yue H (2017) Understanding subtitles by character-level sequence-to-sequence learning. IEEE Trans Ind Inform 13(2):616–624
17. Huang FJ, LeCun Y (2006) Large-scale learning with svm and convolutional for generic object categorization. In: 2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06), vol 1, pp 284–291
18. Lee H, Grosse R, Ranganath R, Ng AY (2009) Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In: Proceedings of the 26th annual international conference on machine learning, ICML'09, New York, NY, USA. ACM, pp 609–616
19. Lee H, Pham PT, Largman Y, Ng AY (2009) Unsupervised feature learning for audio classification using convolutional deep belief networks. Adv Neural Inf Process Syst 22:1096–1104
20. Socher R, Huang EH, Pennin J, Manning CD, Ng AY (2011) Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. Adv Neural Inf Process Syst 24:801–809
21. Glorot X, Bordes A, Bengio Y (2011) Domain adaptation for large-scale sentiment classification: a deep learning approach. In: ICML'11
22. Socher R, Pennington J, Huang EH, Ng AY, Manning CD (2011) Semi-supervised recursive autoencoders for predicting sentiment distributions. In: Proceedings of the conference on empirical methods in natural language processing, EMNLP'11, Stroudsburg, PA, USA. Association for Computational Linguistics, pp 151–161
23. Bordes A, Glorot X, Weston J (2012) Joint learning of words and meaning representations for open-text semantic parsing. In: International conference on artificial intelligence and statistics
24. Heaton JB, Polson NG, Witte JH (2017) Deep learning for finance: deep portfolios. Appl Stoch Models Bus Ind 33(1):3–12
25. Krauss C, Do XA, Huck N (2017) Deep neural networks, gradient-boosted trees, random forests: statistical arbitrage on the S&P 500. Eur J Oper Res 259(2):689–702
26. Chong E, Han C, Park FC (2017) Deep learning networks for stock market analysis and prediction: methodology, data representations, and case studies. Expert Syst Appl 83:187–205
27. Huang G, Huang G-B, Song S, You K (2015) Trends in extreme learning machines: a review. Neural Netw 61:32–48
28. Sun Z-L, Choi T-M, Au K-F, Yu Y (2008) Sales forecasting using extreme learning machine with applications in fashion retailing. Decis Support Syst 46(1):411–419
29. Sun Y, Yuan Y, Wang G (2011) An OS-ELM based distributed ensemble classification framework in P2P networks. Neurocomputing 74(16):2438–2443
30. Handoko SD, Keong KC, Soon OY, Zhang GL, Brusic V Extreme learning machine for predicting HLA-peptide binding. In: Advances in neural networks, ISNN'06
31. Saraswathi S, Sundaram S, Sundararajan N, Zimmermann M, Nilsen-Hamilton Marit (2011) ICGA-PSO-ELM approach for accurate multiclass cancer classification resulting in reduced gene sets in which genes encoding secreted proteins are highly represented. IEEE/ACM Trans Comput Biol Bioinform 8(2):452–463
32. Huang Z, Yu Y, Ye S, Liu H (2014) Extreme learning machine based traffic sign detection. In: 2014 international conference on Multisensor fusion and information integration for intelligent systems (MFI), pp 1–6
33. Kasun LLC, Zhou H, Huang G-B, Vong CM (2013) Representational learning with extreme learning machine for big data. IEEE Intell Syst 28(6):31–34
34. Tang J, Deng C, Huang GB, Zhao B (2015) Compressed-domain ship detection on spaceborne optical image using deep neural network and extreme learning machine. IEEE Trans Geosci Remote Sens 53(3):1174–1185
35. Li X, Xie H, Wang R, Cai Y, Cao J, Wang F, Min H, Deng X (2016) Empirical analysis: stock market prediction via extreme learning machine. Neural Comput Appl 27(1):67–78
36. Seo Y-W, Giampapa J, Sycara K (2004) Financial news analysis for intelligent portfolio management. Ph.D. thesis, Robotics Institute, Carnegie Mellon University
37. Schumaker RP, Chen H (2009) Textual analysis of stock market prediction using breaking financial news: the AZFin text system. ACM Trans Inf Syst 27(2):1–19
38. Li X, Xie H, Song Y, Zhu S, Li Q, Wang FL (2015) Does summarization help stock prediction? A news impact analysis. IEEE Intell Syst 30(03):26–34
39. Van Gestel T, Suykens JAK, Baestaens D-E, Lambrechts A, Lanckriet G, Vandaele B, De Moor B, Vandewalle J (2001) Financial time series prediction using least squares support vector machines within the evidence framework. IEEE Trans Neural Netw 12(4):809–821
40. Cao L, Tay FEH (2001) Financial forecasting using support vector machines. Neural Comput Appl 10(2):184–192
41. Tay FEH, Cao L (2001) Application of support vector machines in financial time series forecasting. Omega 29(4):309–317
42. Tay FEH, Cao L (2002) Modified support vector machines in financial time series forecasting. Neurocomputing 48(1–4):847–861
43. Cao L, Gu Q (2002) Dynamic support vector machines for non-stationary time series forecasting. Intell Data Anal 6(1):67–83
44. Cao L, Tay FEH (2003) Support vector machine with adaptive parameters in financial time series forecasting. IEEE Trans Neural Netw 14(6):1506–1518
45. Huang W, Nakamori Y, Wang S-Y (2005) Forecasting stock market movement direction with support vector machine. Comput Oper Res 32(10):2513–2522
46. Li X, Deng X, Zhu S, Wang F, Xie H (2014) An intelligent market making strategy in algorithmic trading. Front Comput Sci 8(4):596–608
47. Li X, Huang X, Deng X, Zhu S (2014) Enhancing quantitative intra-day stock return prediction by integrating both market news and stock prices information. Neurocomputing 142:228–238
48. Fung GPC, Yu JX, Lu H (2005) The predicting power of textual information on financial markets. IEEE Intell Inform Bull 5(1):1–10
49. Salton G, McGill M (1984) Introduction to modern information retrieval. McGraw-Hill Book Company, New York
50. Li X, Wang C, Dong J, Wang F, Deng X, Zhu S (2011) Improving stock market prediction by integrating both market news and stock prices. Database and expert systems applications.

Lecture notes in computer science, vol 6861. Springer, Berlin, pp 279–293
51. Huang G-B, Zhou H, Ding X, Zhang R (2012) Extreme learning machine for regression and multiclass classification. IEEE Trans Syst Man Cybern Part B 42(2):513–529
52. Maymin PZ (2011) Behavioral finance has come of age. Risk Decis Anal 2(3):125