Knowledge-Based Systems 137 (2017) 138–148

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Intraday prediction of Borsa Istanbul using convolutional neural networks and feature correlations

Hakan Gunduz*, Yusuf Yaslan, Zehra Cataltepe

Computer Engineering Department, Istanbul Technical University, Maslak, Istanbul, Turkey

ARTICLE INFO

Article history:
Received 29 April 2017
Revised 13 September 2017
Accepted 16 September 2017
Available online 19 September 2017

Keywords:
Stock market prediction
Deep learning
Borsa Istanbul
Convolutional neural networks
CNN
Feature selection
Feature correlations

ABSTRACT

Stock market price data have non-linear, noisy and non-stationary structure, and therefore prediction of the price or its direction are both challenging tasks. In this paper, we propose a Convolutional Neural Network (CNN) architecture with a specifically ordered feature set to predict the intraday direction of Borsa Istanbul 100 stocks. Feature set is extracted using different indicators, price and temporal information. Correlations between instances and features are utilized to order the features before they are presented as inputs to the CNN. The proposed classifier is compared with a CNN trained with randomly ordered features and Logistic Regression. Experimental results show that the proposed classifier outperforms both Logistic Regression and CNN that utilizes randomly ordered features. Feature selection methods are also utilized to reduce training time and model complexity.

1. Introduction

Stock market prediction plays a key role in the decision processes of traders and investors. Especially, strategic decisions, such as whether a stock should be bought or sold, are determined according to prediction results. There are several categories of methods for stock market prediction, most using numerical data, such as historical stock prices [1]. Technical analysis and fundamental analysis are two analysis tools that are generally used to predict the future stock price or movement direction. Technical analysis uses past stock prices to predict future price directions, momentums and trends, whereas fundamental analysis uses data about the general economy (e.g., inflation rates, exchange rates, debt ratio, unemployment percentage) and companies (e.g., annual and quarterly reports, income statements) for prediction. Apart from these methods, financial data mining methods that can extract relevant information from financial data, have also been utilized. While Artificial Neural Networks (ANN) and Support Vector Machines (SVM) [2] are the leading algorithms in financial studies, Naive Bayes (NB) [3], Logistic Regression (LR) [4], and K-Nearest Neighbors (KNN) [5] are also widely used due to their simplicity and also ease of interpretation. The success of mentioned algorithms is directly related to the quality of the features selected from the data. Although it is difficult to manually select high quality features that represent temporal and specific properties of the time series data, this can be solved by a deep learning approach. With the help of deep learning, the intrinsic properties of the data can be learned via different abstraction levels of representations and these hierarchical representations can be used as features. Deep learning shows state-of-the-art performances in tasks such as speech recognition, image classification, drug discovery and genetic science [6]. Proven performances in these tasks encourages researchers to use them in stock market prediction [7,8]. Since Deep Neural Networks (DNN) have the ability to learn complex and non-linear relationships from data, it is also a suitable classifier for stock market prediction tasks. By using DNN, it is possible to learn both the relationship between input data and output data, as well as the contribution of input data to the output data.

In this study, we aim to predict the hourly stock price direction of the 100 stocks in Borsa Istanbul (BIST) using different types of classifiers such as Logistic Regression (LR) and Convolutional Neural Network (CNN). 25 technical indicators with different time lags and temporal features are used as inputs into two prediction frameworks. In first framework, filter-based feature selection method, Chi-Square selection, is used to find a common feature subset for all stocks. After common features are indicated, a LR classifier is trained on each stock. In the second framework, instead of using manually selected features, we use feature representations extracted from data and train a CNN classifier. For each stock, a tensor based representation of the historical price informa-

* Corresponding author.
E-mail addresses: hakangunduz@itu.edu.tr (H. Gunduz), yyaslan@itu.edu.tr (Y. Yaslan), cataltepe@itu.edu.tr (Z. Cataltepe).

http://dx.doi.org/10.1016/j.knosys.2017.09.023

tion has been obtained from past 10 days and sorted from highly correlated values to uncorrelated values. Then these sorted tensor representation has been used as an input for CNN. During performance evaluation, because of having skewed class distribution, two frameworks have been compared with Macro-Averaged F-Measure instead of accuracy.

The main contributions of our study can be summarized as follows: First of all, we use a tensor-shaped data in CNN classification that allows us to identify not only the correlations between features, but also the temporal relationship between instances. To the best of our knowledge, this is the first study that uses past and current stock prices with CNN to predict the movement of a stock. Although previously past and current stock prices have been used for the prediction of a stock with Long Short-Term Memory (LSTM) and Recurrent Neural Networks in some studies [9,10], to the best of our knowledge, CNN have not yet been used for prediction in stock market. Experimental results obtained on a 5 years dataset show that the proposed CNN algorithm that use sorted correlated tensor representation outperforms both LR and tensor without unsorted representation. Besides the F-Measure improvement with the proposed method is around 2% on average. Our second contribution is the use of deep learning in Borsa Istanbul prediction. To the best of our knowledge, it is the first study that uses CNN to predict the direction of Borsa Istanbul (BIST) 100 stocks. Although traditional ANN have been used for market-prediction before, CNN have not yet been used to predict the Turkish markets [11].

The remainder of the paper is organized as follows: In the next section, we give an overview of the stock market prediction studies. In Section 3, we describe the dataset used in this study and provide information about the feature selection, and classification methods used. Section 4 gives details of the experimental results. Section 5 concludes the paper.

2. Related work

In this section, we summarize the main approaches used in time series classification. Then, we review some recent studies on stock market prediction that use machine learning techniques and we also cover the related works on Borsa Istanbul. In addition, we also summarize the recent deep learning methods in finance domain.

2.1. The approaches for time series classification

In literature, many approaches have been proposed to solve time series classification. These approaches can be classified into three groups according to the classification scheme: model-based, distance-based and feature-based [12].

In model-based approach, the classes in the time series are assumed to be produced by a model, and the parameters of the model are determined by the training examples of these classes. Thus, different model and model parameters are obtained for different classes. The predicted class for a new time series is determined by comparison of the class models. The autoregressive (AR) model is the simplest method used for this purpose and is widely used for time series classification [13]. The main disadvantage of the AR method is that it assumes that the time series is stationary. On the contrary, Markov Models (MM) and Hidden Markov Models (HMMs) are two models that can be used to model non-stationary time series in model-based approach [14,15].

The second approach in time series classification is distance-based. In this approach, a distance function is defined to measure the similarity or difference between two time series. After finding similarities with the distance function, the time series can be easily classified by K-nearest neighbors (KNN) method. For this reason, the important issue in this approach is how to define the distance function. In literature, the commonly used distance functions to measure the similarity between two time series are Manhattan distance, Euclidean distance and Maximum distance. However, these distance functions have some disadvantages. They only accept time series with the same length and they are sensitive to the time domain distortions [16]. In order to remedy these disadvantages, another distance function called Dynamic Time Warping (DTW) distance is proposed [17].

The third approach used in time series classification is feature-based, where features representing the time series are extracted from the time series data and dimensionality reduction is performed on them. The extraction process of features differs based on the types of time series data [18].

In financial time series data, the raw prices of the markets, the statistical functions calculated from the raw prices, and the technical indicators giving information about the market tendency, volatility and momentum are used as features [19]. In feature-based models, the sampled time series data usually are very noisy and high dimensional [20]. In order to cope with this, dimensionality reduction techniques such as eigenvalue based methods (Principal Component Analysis (PCA), Singular Value Decomposition (SVD)) [21], feature filtering methods (Mutual information, Chi-square, Information gain) [3] can be applied to remove a portion of the noise and reduce the dimensionality. The use of feature reduction has many advantages. The most important of these is to improve the classification performance and to ease the interpretation of the classifier outputs. For this reason, the performance of all feature-based methods depends largely on feature selection and design [22]. In the next subsections, feature-based approaches in stock market prediction and Borsa Istanbul are examined.

2.2. Machine learning for stock market prediction

In literature, several machine learning algorithms have been used for stock market prediction. Generally, to handle non-linearities in financial time series, Neural Networks (NN) [23–25] and Support Vector Machines (SVM) [26,27] have been utilized [2]. In [28], daily maximum and minimum stock prices were predicted with Multi Layer Perceptron (MLP). Technical indicators were used as system inputs and the MLP classifier was employed to decide on the most important indicators for prediction. MLP was also used to forecast the Foreign Exchange Market (FOREX) by using basic technical indicators, such as moving averages, RSI and the standard deviation from several time lags and compared with random selection [23]. Similarly, Radial Basis Function Neural Network (RBFNN) was also trained on several stock market technical variables to predict the movement of the next day's Shanghai Stock Market index [29]. Besides, Principal Component Analysis (PCA) was applied to reduce the dimension of the data feature set. On the other hand, SVM was also used with feature selection to predict the trend of Belgrade stock exchange BELEX15 index [26].

In addition to NN and SVM, other classifiers such as Decision Trees (DT) [30], Random Forests (RF) [31], Logistic Regression (LR) [4] and Naive Bayes (NB) [3] have also been utilized on different stocks using different feature sets.

2.3. Borsa Istanbul studies

Borsa Istanbul (BIST) is a growing stock market and studies that utilize historical index prices, technical indicators and financial news articles have gained great interest recently.

[32] predicted the daily returns of BIST 100 index using Autoregressive models and ANN. [33] predicted the return of the BIST 100 index with Adaptive Neuro Fuzzy Inference System (ANFIS) classifier. Kara et al. [11] made use of technical indicators to predict the

direction of the BIST 100 index and used ANN and SVM. Both classifiers performed well in the prediction process.

In [34], in order to predict the trend of next day price, a subset of features was obtained by removing irrelevant and redundant indicators from the dataset. For this purpose, filter-based feature selection methods were combined, SVM classifier was trained and finally the voting scheme was applied. In order to simulate these processes, a real dataset obtained from the BIST was used with technical and macroeconomic indicators. It was found out that devised feature selection method improves the prediction performance.

In our previous work [3], we proposed a prediction method which uses the analysis of news articles to predict future market movements. We devised a feature selection method called Balanced Mutual Information (BMI) to determine the more relevant features for prediction of the daily BIST100 Index direction. A NB algorithm was used for prediction.

2.4. Deep learning for stock market prediction

Recently, deep learning models have gained great interest and have been used in time-series forecasting such as solar power forecasting [35], energy load forecasting [36] and weather forecasting [37]. Models such as Deep Belief Networks (DBN), Autoencoders, Stacked-Autoencoders (SAE) and Long Short-Term Memory (LSTM) have been employed instead of a simple ANN and SVM in a number of stock market studies.

Peng and Jiang [7] used Deep Neural Networks to predict the direction movement of stocks. Both the price data and the news data were used in the model. The classification was done with Deep Neural Networks using 1024 hidden layer and it was found that the prediction performance was increased by adding the features obtained from news texts to the price features.

Zhu et al. [8] used DBN to design a stock trading decision system. The shares of companies in S&P 500 were used as training and test sets. The system achieved a better result than the buy and hold strategy.

T¿urkmen and Cemgil [38] used SAE, that is an unsupervised deep learning structure, to find the direction of share prices traded on the US Nasdaq Stock Exchange. Technical indicators calculated from the price data were given as input to the deep learning model. Performance of the model was evaluated with accuracy and F-Measure metrics and the proposed model gave the best performance with SVM method.

Chen et al. [9] predicted China stock returns using LSTM network. Two dimensional data points with 30-days-long sequences with 10 features were formed with past China Stock Market data. 3-day earning rate was assigned to data points as class labels. Compared with random prediction method, LSTM model improved the classification performance of stock returns.

Despite superior performances of Convolutional Neural Network (CNN) in speech recognition [39], video classification [40] and sentence classification [41] tasks, it was used rarely in stock market prediction studies[42]. designed a model for event-driven stock market prediction. Firstly, events are extracted from financial news, and represented as dense vectors using word-embeddings. They trained a deep CNN to model both short-term and long-term influences of events on stock price. Their proposed model gave better performance than SVM in S&P 500 index prediction and individual stock prediction.

3. Background

In this section, we give details on the dataset, feature selection methods and classifiers we used in this study.

Table 1
Technical indicators used to extract features.

Indicator	Explanation
OP	Open price
HI	High price
LO	Low price
CL	Close price
ROC	Rate of change
ROCP	Rate of change percentage
%K	Stochastic %K
%D	%D is the moving average of %K
BIAS	x-days bias
MA	x-days moving average
EMA	x-days exponential moving average
MOM	Momentum measures change in stock price over last x days
MACD	x days moving average convergence and divergence
TEMA	Triple exponential moving average
PPO	Percentage price oscillator
CCI	Commodity channel index
WILLR	Larry Williamś %R
RSI	Relative strength index
ULTOSC	Ultimate oscillator
ATR	Average true range
MEDPRICE	Median price
MIDPRICE	Medium price
signL	A signal line is also known as a trigger line
HH	Highest price
LL	Lowest price

3.1. Dataset

The dataset used in this study contains hourly stock prices of the 100 companies listed in BIST 100 Index. Each stock has hourly open, close, high, low prices and volume for 6705 trading hours in between January 2011 and December 2015. The dataset was partitioned into two parts: training set (from January 2011 to December 2014) and testing set (from January 2015 to December 2015). From this data, 25 technical indicators with different time lags (resulting in a total of 75 features) were computed as input features. The decision of which indicators to include was made based on the work of [23,29,34,43]. Details of the technical indicators are summarized in Table 1.

In order to be able to extract as many relevant features as possible, several types of technical indicators were computed. Firstly, for each stock open, high, low and close price values were taken as features. The next features were computed using the indicators given in Table 1. Smoothing indicators such as MA, EMA, and TEMA are indicators that determine potential changes in price information. While ROCP and MOM indicators are influential in finding trends in the series, whereas oscillator indicators such as %K , %D (moving average of %K), WILLR, BIAS, RSI, CCI and PPO are good for determining oversold or overbought market conditions. The ATR indicator can be applied to prices to assess the volatility of a stock. High ATR values indicate strong bi-directional movement that can be used in stop-loss decisions. MACD and ULTOSC are indicators that should be used together to give reliable signals related to market momentum. LL and HH show the support and resistance levels respectively. These indicators can indicate near-low and near-high price levels according to the trend of the market [43]. Since each indicator covers different types of information, their combination can yield better results.

After the indicator computation, we had a dataset with 6705 instances and 75 features for each stock. Besides the features of each stocks, we also added some temporal features such as the hour of the day, the day of the week, the day of the month and season information. Except for the day of the month feature, which has a value between 1 and 31, other attributes were created with 1-of-K representation. After adding the temporal features, we had totally

94 features for each trading hour and each stock. After forming feature vectors, we applied min-max normalization on each feature shown in Eq. (1):

$$z_i = \frac{x_i - \min(x)}{\max(x) - \min(x)} \tag{1}$$

In Eq. (1), $x = (x_1, \ldots, x_n)$ denotes the feature vector to be normalized and z_i denotes the normalized value of x_i. In order to not to learn from the test portion of the time series data, we computed the normalization parameters, $\min(x)$ and $\max(x)$ on the training sets and then used these parameters for the normalization on the test sets.

Class labels indicate the movements of the stock prices and they are computed using the close prices of the stocks. Let $c(t)$ and $c(t-1)$ denote the close price for a stock in hour t and the previous trading hour $t-1$. The class label, $r(t)$, for tth hour is computed as:

$$r(t) = \begin{cases} 1, & \text{if} \dfrac{c(t) - c(t-1)}{c(t-1)} \geq 0.002 \\ -1, & \text{otherwise} \end{cases} \tag{2}$$

The threshold value (0.002) used in determining the class label is decided by taking into account of the exchange commission rate in Borsa Istanbul. After class labels are computed, they are assigned to their respective feature vectors.

3.2. Feature selection

Feature selection algorithms find a set of relevant features for the specific problem set. Removal of the redundant, noisy and irrelevant features from feature space leads to simpler classifiers that can have better generalization on an unseen data. The reduction in the size of the feature set, especially for high-dimensional data, results in faster classification and models that are easier to interpret and explain. In order to eliminate the noise and irrelevant features we also applied Chi-Square feature selection method on our datasets.

3.2.1. Chi-square (χ^2)

Chi-square selection is commonly used in machine learning to measure the dependence between a feature and the class label. Let c_i denote the existence and $\bar{c}_i$ denote the non-existence of the class $i \in \{0, 1\}$. Let $P_j(1, c_i)$ and $P_j(0, \bar{c}_i)$ denote the probability of feature j and class i co-occurring and the probability of both the feature and class not occurring together respectively. Let us define $P_j(1, \bar{c}_i)$ and $P_j(0, c_i)$ similarly. Then the Chi-square score for feature $j \in \{1, \ldots, m\}$ and class $i \in \{0, 1\}$ is compute as:

$$\chi^2(w_j, c_i) = \frac{N[P_j(1, c_i)P_j(0, \bar{c}_i) - P_j(1, \bar{c}_i)P_j(0, c_i)]^2}{P_j(1)P_j(0)P(c_i)P(\bar{c}_i)} \tag{3}$$

where N is the total number of instances and note that if feature w_j and the class i are correlated or inversely correlated, their χ^2 score would be high. If they are purely independent from each other, then, the score would be zero [3].

3.3. Classifiers

In our study, the direction of the closing price of BIST 100 stocks was predicted with Logistic Regression and Convolutional Neural Network classifiers.

3.3.1. Logistic regression

The first classifier we used was Logistic Regression (LR). In our study we used LR to evaluate the relationship between binary class labels (-1 or $+1$) and multiple features (technical indicators). The LR model gave us the probability of following trading hour will be decided as "Up":$+1$ or "Down":-1. In our case, we determined the threshold value as 0.5 and we assigned class label as "Up" if the probability exceeded the threshold.

LR estimates the probability of output as follows:

$$p(y|x_1, \ldots, x_n) = \frac{e^{w_0 + w_1x_1 + w_2x_2 + \ldots + w_nx_n}}{1 + e^{w_0 + w_1x_1 + w_2x_2 + \ldots + w_nx_n}} \tag{4}$$

In this equation, y is the class label and $p(y|x_1, \ldots, x_n)$ is the posterior probability of the movement direction of a trading hour given the n features $(x_1, \ldots, x_n)$. In order to compute the model parameters $(w_0, \ldots, w_n)$, the maximum likelihood method is used [44].

3.3.2. Convolutional neural network

Convolutional Neural Network (CNN) is basically composed of multiple layers where the convolution operation is performed on. The main difference between ANN and CNN is the connection between the layers. In CNN, each local part (receptive field) of the inputs is connected to only one neuron while the inputs in ANN are fully connected to the neurons in the next layer. In each layer of CNN, convolution operation is done by applying different-sized filters on inputs. After convolution process, outputs of convolution layers are passed through the activation function. Then, pooling layers are employed in subsampling from activated outputs. By the process of pooling, input data of different dimensions can be processed by the network.

CNN's prominent properties are location invariance and compositionality. For example, if an object recognition (e.g., dog) task is done using CNN, it does not matter where the object is located in the image, since the filters used will process the entire image. If the image contains the dog object, it can be revealed by the pooling process. Pooling is the solution to rotation and scaling in images and it brings location invariance property. Filters applied to the input data in CNN convert the low-level attributes of the local part of the data into a high-level attribute representation on each layer. This shows the compositionality property of the CNN [45].

However, CNN has several hyper parameters that are filter size, stride, and pooling type. The filter size indicates whether the filter is applied to each element of the input matrix or to the region of the matrix. The stride shows how many steps are to be taken in each step of the filtering. The pooling type indicates whether the pooling process will be applied to each filter region or to the feature map that is the result of the filtering process [46].

3.3.3. Feature-based models vs. convolutional neural networks

Time series classification models are generally feature-based which highly depend on the extracted features from data. However, it is generally difficult to obtain relevant and key features for capturing the latent attributes of time series data. Extraction of relevant features is also time-consuming and needs domain expertise [47]. Moreover, the noise in the time series data causes problems in extraction of useful information. Shallow methods that contain only a few number of non-linear transformations do not have enough capability to model complex, noisy and high dimensional time series data [48].

Recently, Deep Neural Networks (DNN) and in particular Convolutional Neural Networks (CNN) have been applied to time series data [39,40] due to their strong generalization and noise tolerance performance [12]. The capability of extracting abstract features and hidden non-linear relations from data without the need for human expertise, makes CNN a strong alternative to existing feature-based models [49]. CNN can explore and extract the intrinsic properties of the input time series using convolution and pooling operations [50].

In several time series studies, CNN is used as a feature extractor combined with machine learning models [20,51]. In these studies,

Table 2
Confusion matrix for two-class classification.

Actual/Predicted as	Positive	Negative
Positive	*tp*	*fn*
Negative	*fp*	*tn*

the feature learning ability of CNN generates robust features without the human intervention. CNN can have superior discriminative power when multiple convolutional layers are stacked on top of each other. Also, it can achieve invariance to noise and reduce the dimensionality of the feature space by using the pooling operation [48]. CNN also can be employed as a classification framework [50,52]. This framework is an entirely end-to-end neural network in which the input data is the time series to be predicted and the output is its label/value.

3.4. Evaluation metrics

Evaluation metrics are needed to measure and compare the predictability of classifiers.

Accuracy is one of the most commonly used metrics. However, for an unbalanced dataset, the accuracy can only be high if only the majority label is assigned for any given instance. For this reason, most classification methods can assign the majority label as output by default.

Evaluation metrics such as precision, recall and F-measure can measure how well a classifier can distinguish among different classes, even in case of class imbalance.

Let the confusion matrix as in Table 2 store the counts of correctly and incorrectly classified instances per class for a binary classification. In the confusion matrix, *tp, fp, fn* and *tn* show true positive (tp), false positive (fp), false negative (fn) and true negative (tn) counts, respectively. Based on these counts, precision, recall and F-Measure are computed as:

$$\textbf{precision} = \frac{tp}{tp + fp} \tag{5}$$

$$\textbf{recall} = \frac{tp}{tp + fn} \tag{6}$$

$$\textbf{F} - \textbf{Measure} = \frac{2 * precision * recall}{precision + recall} \tag{7}$$

These three measures are defined according to the confusion matrix values of a two-class problem. In order to evaluate the classifier performance for each class, we used the Macro-Averaged F-Measure, by taking the mean of the F-Measure values for positive and negative class [53].

4. Experiments

As we mentioned in previous sections, we had two different frameworks in classifications. In order to compare the performance of our classification frameworks, we used two time-series baseline methods: Random and Naive approaches. Due to skewed class distribution (ratio is approximately 2:1) in our data, the classification performances of the proposed methods and time-series baseline methods were evaluated using Macro-Averaged (MA) F-Measure metric.

In random approach we assign class labels to stock prices randomly based on class distributions of price directions. In this approach, the prior probability of each class are found by using class distributions in the training data. Then, the cumulative probabilities are calculated according to found probability values. Finally, random values are generated between 0 and 1 and the class labels are determined according to the range in which these values fall within the cumulative probabilities.

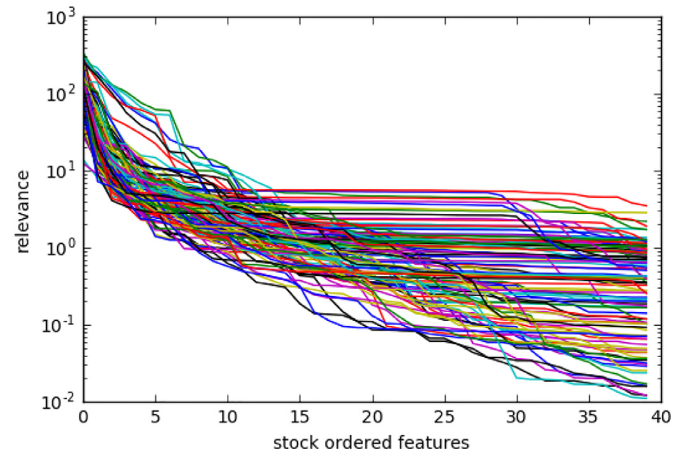

Fig. 1. Ordered relevances of features for each instrument. The relevances for features 41 and beyond were less than are not shown here.

Table 3
Relevant features selected by Chi-square.

Indicators	Counts
WILLR-8	100
Hour8	100
Hour2	88
WILLR-16	81
Hour7	81
Hour3	80
Hour4	72
Hour6	71
Hour5	70
CCI-8	66
WILLR-24	63
WILLR-32	55
Season3	52
CCI-2	50

Second baseline prediction is obtained using a naive approach where the value of current observation t is equal to the previous observed label of $t-1$. The classification results of these approaches are given in Table 5 for each stock. Note that the F-Measures of the random model is better than the naive approach and the best performance is around 51%.

The second experiments are obtained using individual stock features with L2 regularized Logistic Regression (LR). Each stock has 94 features with 4765 training and 1940 test instances. The results are shown at Table 6. In order to compare the overall performance of the LR with the baseline approaches, we took the mean of MA F-Measure values of all stocks. The results show that LR classifier (0.547 MA F-Measure rate) shows better performance than Naive (0.465) and Random (0.499) approaches in terms of mean MA F-Measure.

After getting results with all features, we employed feature selection to select the relevant features. In order to find the feature relevances for each stock, Chi-Square feature selection was used. High-to-low ranked relevance scores for each stock were plotted on top of each other in Fig. 1. The figure shows that compared to all the features, the first 20 features of each stock has a higher relevance score. Therefore, we selected these 20 features for each stock and grouped them according to the selected indicators and how many stocks that indicators belong to. Features selected by at least 50 stocks and their counts are shown in Table 3. According to this table, the trading hour information, except the opening

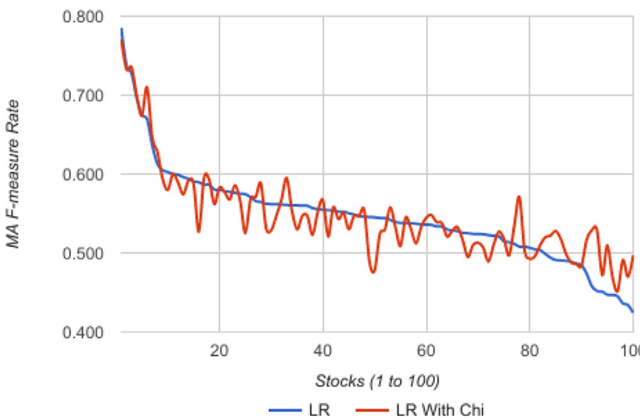

Fig. 2. Comparison of classification performances: LR vs. LR with selection.

hour was informative. Among all the indicators used, WILLR and CCI were selected.

In order to evaluate the performance of Chi-Square selection, we used the features shown in Table 3 and trained LR classifier for each stock. The results can be seen in Table 6. In these classifications, we got 0.545 mean MA F-Measure rate using only 14 features. We did almost the same performance with the result of without feature selection. LR classifier with Chi-Square selection again showed better performance than Naive and Random approaches in terms of mean MA F-Measure.

In order to compare the performances of Chi-Square selection with no selection in LR classification, the classification results are plotted in Fig. 2. Fig. 2 reveals that the classification results obtained with Chi-Square selection are higher, especially when stocks had 0.50 or less MA F-measure rate.

The next experiment was performed with Convolutional Neural Network classifier (CNN). Details of proposed CNN classifier are explained in next subsection.

4.1. Proposed CNN classifier

Our CNN classifier has a total of 7 layers with 1 input layer, 2 parallel convolution layers, 1 merge layer, 1 convolution layer, 1 fully-connected (dense) layer and 1 output layer. The graphical representation of the proposed CNN architecture is given in Fig. 3. The overall architecture of the proposed CNN has two main sequential stages: parallel layer, and merge layer:

Each part (branch) in parallel layer consists of convolution and pooling layers which are used to extract deep features from input data separately for each branch. All the outputs of convolution layers will pass through a max pooling of size 2 by 2. In the parallel layer, the convolution for different branches are independent from each other. Parallel layer generates multiple time series representations of different time scales and allows the input data to be viewed from different perspectives. In the merge layer, all extracted features from parallel layers are concatenated. One more convolutional layer followed by max pooling, dense layer, and an output (a soft-max) layer is utilized to generate the final output. This network is a complete classification framework and all weights are trained jointly using the back propagation algorithm.

In this study, we used the same analogy as in image classification, where the spatial relationship between each pixel regions have been learned by CNN. Instead of extracting spatial relations (as in image classification task), we aimed to extract the relationship between current trading hour and previous trading hours while considering correlations between features. In order to do this, we transformed each instance into 2D-matrix. The transformation of each instance was done by combining instance at time t with n previous instances ($t-1$, $t-2$, ..., $t-n$). The class label of hour t, $r(t)$, is predicted by considering the instances from hour t to hour $t-80$ which means we were considering previous 10 trading days (Each trading day has 8 trading hours). Since CNN attaches importance to spatial relationships between the features, we need to change the order of features in data transformation to be able to discover the relationships between the features. In order to this, we computed the feature correlations on each stock and took the average of the computed correlations. Then, we applied hierarchical agglomerative clustering [54] on the mean correlation matrix and clustered the features according to the values of the feature correlations. The positions of the features in the 2D-matrix were rearranged considering the order of the clustered features in the dendrogram. So that the input data was transformed by taking into account both the use of the previous instances and the correlations between the features.

Table 4
CNN parameters.

Parameter: {Value}
Filter size: {8}
Size of max-pooling: {(2, 2)}
Optimizer: {Adadelta}
Activation function: {RELU}
#Epochs: {200}
Batch size: {32}
Dropout rate: {0.3}

With our data transformation, we considered two issues in multivariate time series classification. The first issue is temporal information between the instances. Since our time series data have a temporal structure, we can not treat each instance as an independent feature vector, because this will neglect the temporal information between instances [48]. This issue can be solved by remembering the past instances while training the model. To do this, we use a tensor-shaped data in our proposed CNN that allows us to identify the temporal relationship between instances. So, temporal consistency is included in our model. The second issue is the relationships between features. In some CNN studies, multivariate time series are split into univariate ones for feature learning [51]. However, this approach does not have ability to detect the relationships between different univariate time series. In order to mine the relationships between our features, we use our data in multivariate fashion and only modify the order of features. Instead of using random ordered features in our tensor-shaped data, the positions of the features are rearranged considering the order of the clustered feature correlations.

Since the dataset used in the study is relatively small for deep learning, the parameters of the proposed CNN architecture has been utilized with cross-validation procedure. Because of long training times in learning, we chose 20 out of 100 stocks randomly and performed a grid search with 5 times cross-validation to determine optimal hyper parameters (optimization function, activation function and dropout rate). After finding the best parameter set for each 20 stock, the common parameter set for all stocks was determined based on the majority of the values of these parameters. We trained our CNN classifiers using KERAS [55] package with defined hyper parameters shown in Table 4. In classification process, three different techniques have been used to prevent the over-fitting in training:

- L2 regularizer is the first technique to prevent the over-fitting by adding the squared magnitude of all weights to the objective function to penalize extremely large weights [41].
- Dropout is another technique used to solve the problem of over-fitting. In the training phase, Dropout only keeps the neuron active with some probability value, p, or otherwise sets it

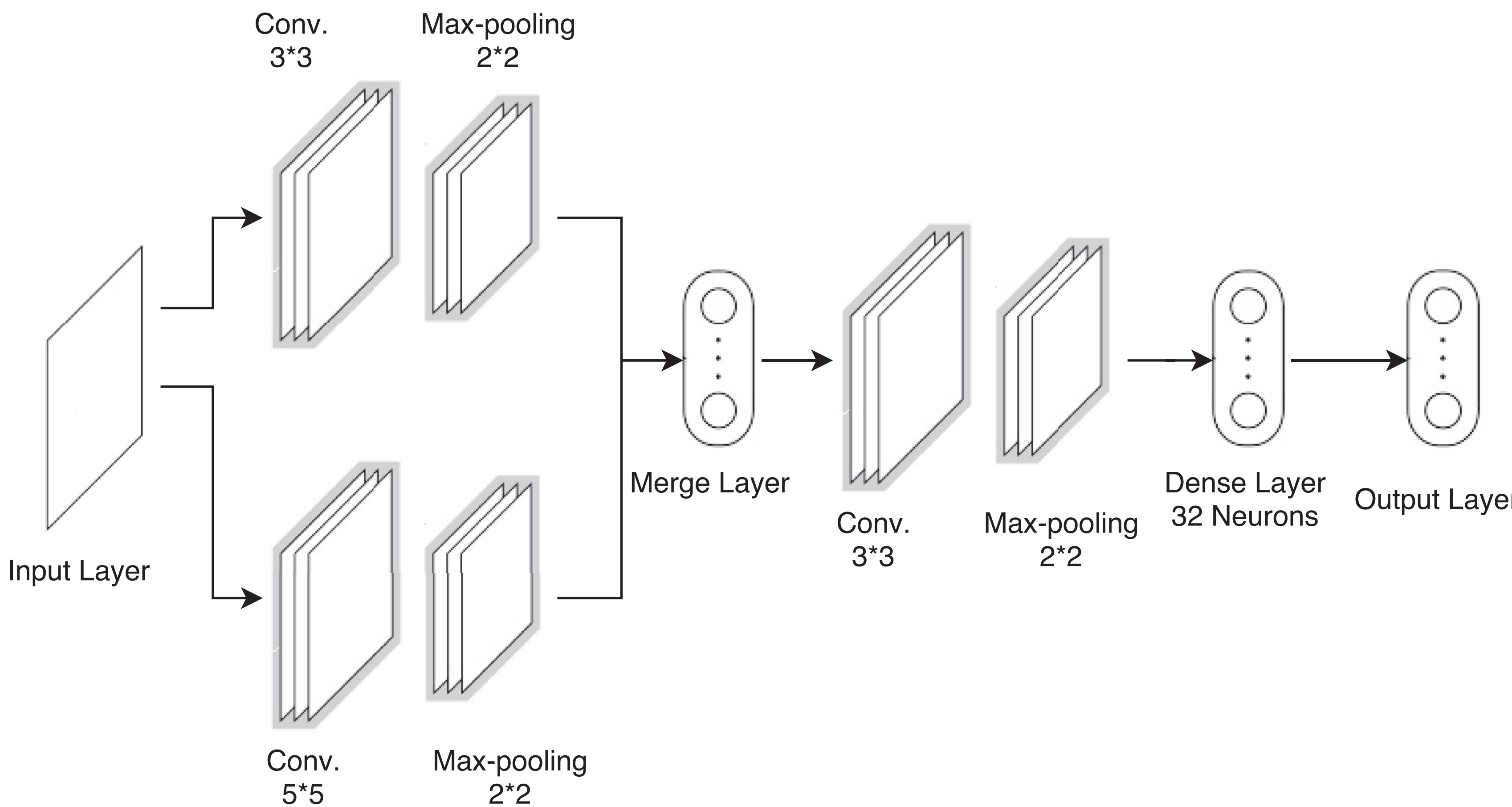

Fig. 3. Graphical representation of proposed CNN.

Table 5
Stocks and the MA F-measure values obtained by Naive and Random approaches.

Stocks	*Random*	*Naive*	*Stocks*	*Random*	*Naive*	*Stocks*	*Random*	*Naive*	*Stocks*	*Random*	*Naive*
GOODY	0.503	**0.524**	AFYON	**0.513**	0.487	YAZIC	**0.506**	0.468	TSKB	**0.509**	0.444
TUPRS	0.507	**0.520**	HALKB	**0.500**	0.485	GOZDE	**0.478**	0.466	ZOREN	**0.495**	0.444
GOLTS	0.513	**0.516**	YKBNK	**0.503**	0.483	SISE	**0.495**	0.466	ECILC	**0.498**	0.444
GUBRF	0.492	**0.514**	EGEEN	0.475	**0.483**	TRKCM	**0.531**	0.465	CIMSA	**0.496**	0.442
AKBNK	0.503	**0.509**	MGROS	**0.505**	0.482	AYGAZ	**0.480**	0.464	KIPA	**0.511**	0.441
SODA	0.483	**0.507**	ECZYT	**0.501**	0.482	AEFES	**0.531**	0.464	IZMDC	**0.524**	0.441
THYAO	**0.520**	0.507	BRISA	**0.510**	0.482	LOGO	**0.495**	0.462	AKSEN	**0.512**	0.440
FENER	0.491	**0.503**	BRSAN	**0.493**	0.481	ASELS	**0.503**	0.462	NUGYO	**0.488**	0.436
DOCO	0.498	**0.502**	TKFEN	**0.482**	0.480	GLYHO	**0.487**	0.462	NTTUR	**0.499**	0.433
ENKAI	**0.516**	0.502	ALARK	**0.492**	0.479	TAVHL	**0.488**	0.461	ISGYO	**0.477**	0.433
ISCTR	0.493	**0.500**	ULKER	**0.493**	0.479	ARCLK	**0.496**	0.461	ASUZU	**0.490**	0.431
KONYA	**0.504**	0.500	SAHOL	0.472	**0.478**	ERBOS	**0.496**	0.459	KARSN	**0.514**	0.429
GARAN	0.496	**0.499**	FROTO	**0.505**	0.477	PRKME	**0.493**	0.458	BOSSA	**0.505**	0.429
TATGD	0.485	**0.498**	VESTL	**0.496**	0.476	KRDMD	**0.492**	0.457	ANACM	**0.484**	0.427
OTKAR	**0.508**	0.496	NETAS	**0.518**	0.476	KCHOL	**0.506**	0.456	SKBNK	**0.505**	0.426
CLEBI	**0.507**	0.496	KARTN	**0.510**	0.475	GSRAY	**0.481**	0.456	GSDHO	**0.496**	0.421
ALCTL	**0.534**	0.495	EKGYO	**0.500**	0.475	IPEKE	**0.505**	0.455	TSPOR	**0.504**	0.419
ALGYO	0.488	**0.495**	KOZAA	**0.497**	0.474	TCELL	**0.505**	0.455	METRO	**0.507**	0.413
BIMAS	0.480	**0.495**	DOAS	**0.485**	0.474	VESBE	**0.514**	0.454	CEMAS	**0.507**	0.410
KOZAL	**0.509**	0.495	AKSA	**0.499**	0.474	BIZIM	**0.479**	0.454	ALBRK	**0.510**	0.400
CCOLA	**0.497**	0.494	DEVA	**0.497**	0.473	AYEN	**0.493**	0.453	IHEVA	**0.510**	0.398
VAKBN	**0.493**	0.493	TOASO	**0.511**	0.472	TRGYO	**0.524**	0.451	AKENR	**0.506**	0.397
TTKOM	0.488	**0.492**	EREGL	**0.495**	0.470	VKGYO	**0.499**	0.449	HURGZ	**0.488**	0.395
TTRAK	**0.513**	0.490	PETKM	**0.504**	0.469	BJKAS	**0.482**	0.446	DOHOL	**0.509**	0.392
KORDS	0.482	**0.490**	BAGFS	**0.489**	0.469	TRCAS	**0.482**	0.445	IHLAS	**0.498**	0.386

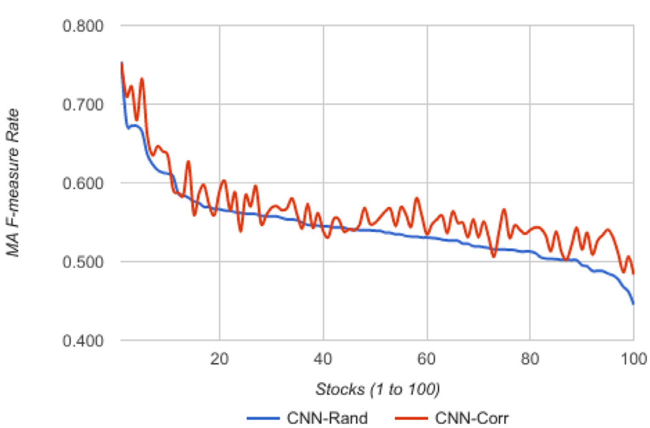

Fig. 4. Comparison of classification performances: CNN-Rand vs. CNN-Corr.

to zero. Hence, Dropout can be thought as a neural network sampling within the full neural network, and the weights are updated only for the sampled network based on the input data [56].

- The third technique is early stopping which is used to find the optimal number of epochs [57]. While using too many epochs can lead to over-fitting, under-fitting can occur with too few epochs. With the help of early stooping, we can control over-fitting status at the end of every *N* epochs with validation error. If the validation error of the current model has less error than the previous best saved model, this model can be stored as the best model. This process goes on until the current model has higher validation error than the saved models.

In order to examine the effects of the feature correlations, we trained two types of CNN with different feature orderings. While the first CNN, named as CNN-Rand, used random-ordered features in input data whereas the second CNN, named as CNN-Corr, considered the feature correlations to rearrange the order of the features. The classification results of CNN-Rand and CNN-Corr classifiers for 100 stocks (Table 6) are plotted in Fig. 4. The results are sorted by CNN-Rand in descending order. The results show that for the most predictions (87 of 100 stocks), CNN-Corr had better MA F-Measure values than CNN-Rand. The same results also can be seen when looking at the mean MA F-Measure rates. CNN-Corr had a mean MA F-Measure rate of 0.563 while CNN-Rand achieved 0.544 mean MA F-Measure. We also compared CNN-Corr and CNN-Rand results with our baseline methods. Both CNN-Corr and CNN-Rand classifiers outperforms Naive (0.465) and Random (0.499) methods in terms of mean MA F-Measure rates.

To sum up the results mentioned above, CNN-Corr classifier achieves the best performance among all classifiers in terms of mean MA F-Measure rate. Using feature correlations in this CNN increases the classification performance nearly 2% according to CNN-Rand classifier. LR, with or without feature selection, performed almost the same as CNN-Rand, while feature selection with LR achieved this success using only 14 commonly-selected features. We also compared the algorithms between each other using a normalized F-Measure metric to grasp the general tendencies of the stocks. In the next subsection we detail the analysis.

4.2. Normalized F-measure of classifiers

In our experimental setup, we had four different classifiers: LR w/o selection, LR with selection, CNN-Rand and CNN-Corr. In order to decide on the performance of a classifier, we tried to find how efficient each classifier is, i.e., can it perform the best for each stock?.

In order to find the relative MA F-Measure for each classifier, the MA F-Measure rate found for each classifier was divided by the maximum F-Measure rate obtained for the same stock. After this normalization process, relative MA F-Measure rate for each classifier and for each stock were found. This allows the comparison of classifiers and MA F-measure values for different stocks. In Fig. 5, histograms were created according to the relative MA F-Measure values of each classifier.

According to Fig. 5, CNN-Corr classifier outperforms other 3 classifiers in terms of its relative MA F-Measure. In CNN-Corr classification, great majority of the stocks have relative MA F-measure values between 0.95 and 1.00. This is also seen when the best MA F-Measure scores for the stocks are examined. CNN-Corr had 54 best MA F-Measure scores in 100 stocks whereas LR w/o selection,

Table 6
Stocks and the MA F-measure values obtained by each classification method. Stocks are ordered according to LR performance.

Stocks	*LR*	*LRWithFS*	*CNN-Rand*	*CNN-Corr*	*Stocks*	*LR*	*LRWithFS*	*CNN-Rand*	*CNN-Corr*
IHLAS	**0.785**	0.771	0.754	0.753	TTKOM	**0.544**	0.524	0.488	0.534
CEMAS	**0.738**	0.733	0.673	0.722	ECZYT	**0.544**	0.530	0.462	0.506
IHEVA	0.727	**0.735**	0.665	0.732	THYAO	0.542	**0.558**	0.540	0.546
HURGZ	0.696	**0.699**	0.672	0.679	YKBNK	0.539	0.533	0.513	**0.540**
METRO	**0.675**	0.674	0.637	0.661	CIMSA	0.538	0.509	0.543	**0.554**
DOHOL	0.670	0.710	0.676	**0.710**	PRKME	0.538	0.545	0.537	**0.564**
BRSAN	0.637	**0.646**	0.624	0.635	NUGYO	0.538	0.531	**0.558**	0.548
GSDHO	0.613	0.628	0.613	**0.640**	AYEN	0.537	0.512	0.543	**0.554**
NTTUR	0.605	0.594	0.581	**0.627**	MGROS	**0.536**	0.533	0.530	0.535
AKENR	0.603	0.580	0.616	**0.647**	CLEBI	0.536	**0.545**	0.485	0.541
IPEKE	0.600	**0.600**	0.554	0.580	BJKAS	0.536	0.548	0.566	**0.590**
IZMDC	**0.599**	0.589	0.560	0.596	ASELS	0.534	0.539	0.533	**0.560**
ALBRK	0.596	0.574	0.612	**0.634**	BAGFS	0.533	0.538	**0.549**	0.542
KONYA	**0.594**	0.592	0.561	0.570	KOZAL	0.530	0.521	0.506	**0.542**
BIZIM	**0.591**	0.587	0.545	0.539	TSKB	0.529	0.528	0.546	**0.573**
ECILC	**0.590**	0.527	0.569	0.571	AKBNK	0.528	0.533	0.539	**0.549**
BRISA	0.587	**0.595**	0.557	0.570	TKFEN	0.526	0.517	**0.562**	0.538
GOODY	0.586	**0.593**	0.555	0.565	PETKM	0.525	0.495	**0.543**	0.538
TSPOR	0.580	0.562	0.532	**0.580**	AKSA	0.524	0.510	0.520	**0.554**
SKBNK	0.580	0.583	**0.608**	0.592	VAKBN	0.524	0.513	0.517	**0.530**
GUBRF	**0.579**	0.577	0.482	0.531	EREGL	0.524	0.507	0.511	**0.543**
KIPA	0.577	0.568	0.574	**0.586**	GLYHO	0.523	0.489	0.539	**0.556**
TRCAS	0.576	**0.586**	0.567	0.559	TRGYO	0.522	0.510	0.515	**0.539**
VKGYO	0.575	0.566	0.569	**0.597**	GARAN	0.521	0.527	0.515	**0.530**
VESBE	**0.574**	0.525	0.540	0.568	AKSEN	0.515	0.518	0.523	**0.531**
KARTN	0.571	0.568	**0.576**	0.560	SISE	0.514	0.497	0.504	**0.533**
VESTL	0.566	**0.571**	0.537	0.567	DOCO	0.512	0.534	**0.541**	0.540
ALCTL	0.565	**0.588**	0.584	0.585	GOLTS	0.508	**0.571**	0.541	0.541
KORDS	**0.562**	0.532	0.535	0.545	YAZIC	0.508	0.503	0.530	**0.547**
KRDMD	**0.562**	0.529	0.529	0.554	EKGYO	0.507	0.493	0.515	**0.566**
ALARK	**0.562**	0.547	0.544	0.531	KARSN	0.505	0.496	0.527	**0.564**
KOZAA	0.561	**0.568**	0.564	0.565	AYGAZ	0.504	0.510	0.527	**0.536**
AFYON	0.561	**0.595**	0.563	0.588	LOGO	0.499	0.520	0.528	**0.558**
NETAS	0.561	0.554	0.561	**0.584**	DOAS	0.494	0.522	0.527	**0.549**
DEVA	**0.560**	0.530	0.468	0.486	ERBOS	0.491	0.528	0.531	**0.560**
ISGYO	**0.560**	0.547	0.523	0.549	KCHOL	0.491	0.518	0.515	**0.546**
EGEEN	**0.560**	0.547	0.502	0.543	CCOLA	0.490	0.499	0.513	**0.535**
ZOREN	0.557	0.523	0.565	**0.602**	TATGD	0.489	0.488	0.488	**0.526**
HALKB	0.556	0.552	0.534	**0.569**	ULKER	0.486	0.486	0.532	**0.544**
GOZDE	0.555	0.567	0.554	**0.568**	TCELL	0.485	0.484	0.519	**0.531**
BOSSA	0.554	0.521	0.552	**0.559**	ENKAI	0.474	0.517	0.494	**0.537**
ALGYO	0.554	0.558	0.545	**0.562**	TTRAK	0.458	0.529	0.503	**0.538**
ANACM	0.552	0.543	0.586	**0.588**	OTKAR	0.452	**0.529**	0.488	0.509
FENER	**0.552**	0.551	0.540	0.548	BIMAS	0.451	0.472	0.502	**0.513**
TRKCM	**0.550**	0.530	0.502	0.521	ARCLK	0.447	0.510	0.513	**0.540**
ASUZU	0.549	0.547	0.558	**0.559**	TAVHL	0.447	0.469	0.502	**0.502**
GSRAY	0.547	0.547	0.558	**0.568**	SAHOL	0.445	0.452	0.503	**0.513**
ISCTR	0.546	**0.554**	0.546	0.543	FROTO	0.436	0.491	**0.515**	0.506
SODA	**0.546**	0.489	0.477	0.511	TOASO	0.434	0.470	0.495	**0.515**
AEFES	0.545	0.478	0.518	**0.551**	TUPRS	0.424	**0.497**	0.445	0.483

LR with selection and CNN-Rand classifiers had 21, 17 and 8 best F-measure scores respectively.

5. Conclusions

In this study, we predicted the hourly movements of 100 stocks in Borsa Istanbul using different types of technical indicators and temporal features. Class labels were assigned using each stock hourly close prices and the movements in stock prices were indicated by these labels. Logistic Regression (LR), and Convolutional Neural Network (CNN) classifiers were trained and its performance was measured in terms of Macro-Averaged (MA) F-Measure metric.

Experimental result are obtained on 4 years training and 1 year test set. In the first experimental setup, Chi-square feature selection is used to find a common feature subset for all BIST 100 stocks. After common features are determined, a LR classifier is trained for each stock. Experimental results show that the results obtained with feature selection (0.545 MA F-Measurement rate) are similar to those obtained without selection.

In the second experimental framework, instead of using manual selected features, we used feature representations extracted from data with CNN. We extracted the relationship between current trading hour and previous trading hours while considering the correlations between the features. Each instance is transformed into 2D-matrix by combining previous instances and considering the features correlations. We examined the effects of feature correlations by training two types of CNN with different feature orderings. While we performed 0.544 mean MA F-Measure rate with random-ordered features (CNN-Rand), we achieved higher F-Measure rate (0.563) with the reordered features according to clustered feature correlations (CNN-Corr). The proposed algorithm also outperformed LR with and without feature selection.

Upon examination of the relative MA F-Measure scores of each classifier, it was found out that CNN-Corr achieved better relative MA F-Measure values than other three classifiers. CNN-Corr was the leading algorithm with the best MA F-Measure scores in 54 of 100 stocks.

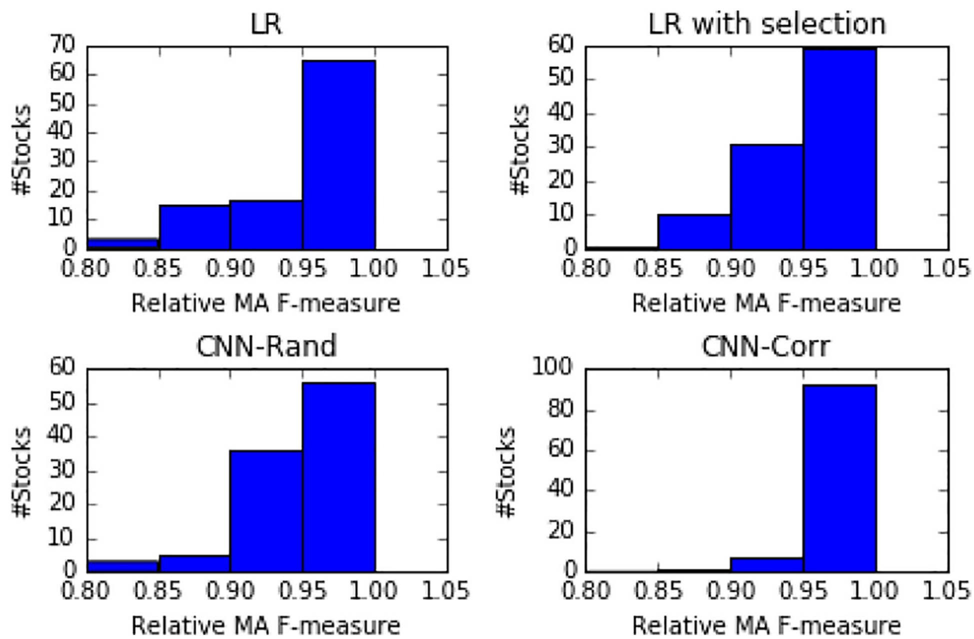

Fig. 5. Histograms of relative MA F-Measure scores belonging to 4 classifiers.

The main advantages of the CNN-Corr classifier introduced in this work with respect to the feature based models in the previous studies [58–60] can be summarized as follows:

- Unlike recent financial prediction studies [58,59] that include feature selection and classification steps separately, CNN-Corr is an entire classification framework that combines feature selection and classification steps.
- The proposed CNN structure uses parallel representations, particularly using different convolution sizes. This allows to focus the attention within the data by exploiting different points of view and perspectives and has robustness to shifting and scaling invariance [20].
- At each time step, we consider up to 94 features, then apply an agglomerative clustering. This approach allows to correctly define the spatial order of such features in our data and to mine the interrelationships between them.
- Most financial studies use accuracy which can be a misleading metric in case of imbalanced class distribution [60]. Inspired by study [61], we use F-measure metric to analyze skewed dataset classification results. F-measure seems to be very promising and efficient from the point of view of unbalanced dataset.

Future works

Temporal relationships and feature interrelations are generic features of multivariate time series data. Since our proposed CNN has ability to extract feature representations with its generic structure, it can be applied on several different multivariate time series such as wind, water and electricity. In the future, we plan to use our CNN classifier in electricity consumption prediction [62] and wind speed prediction [63].

With the help of parallel convolutional layers in our proposed CNN, different types of features simultaneously can be fed into the network as inputs. This bring the opportunity to process the multimodal data in time series prediction. In the future, we plan to use different types of data such as financial news and price data in stock market prediction. Also, we plan to do a further analysis of feature dependencies and produce stock networks based not only on the features but also on the labels. We believe that this multi-label approach could allow detection of causal dependencies between classes.

References

[1] D. Marcek, Stock price forecasting: statistical, classical and fuzzy neural network approach, Lect. Notes Comput. Sci. 3131 (2004) 41–48.
[2] G.S. Atsalakis, K.P. Valavanis, Surveying stock market forecasting techniques—part II: soft computing methods, Expert Syst. Appl. 36 (3) (2009) 5932–5941.
[3] H. Gunduz, Z. Cataltepe, Borsa Istanbul (BIST) daily prediction using financial news and balanced feature selection, Expert Syst. Appl. 42 (22) (2015) 9001–9011.
[4] A. Dutta, G. Bandopadhyay, S. Sengupta, Prediction of stock performance in indian stock market using logistic regression, Int. J. Bus. Inf. 7 (1) (2015).
[5] I. Vainionpää, S. Davidsson, Stock market prediction using the k nearest neighbours algorithm and a comparison with the moving average formula (2014). http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-157696.
[6] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.
[7] Y. Peng, H. Jiang, Leverage financial news to predict stock price movements using word embeddings and deep neural networks, (2015). arXiv:1506.07220.
[8] C. Zhu, J. Yin, Q. Li, A stock decision support system based on DBNs, J. Comput. Inf. Syst. 10 (2) (2014) 883–893.
[9] K. Chen, Y. Zhou, F. Dai, A LSTM-based method for stock returns prediction: a case study of china stock market, in: Proceedings of the IEEE International Conference on Big Data (Big Data), IEEE, 2015, pp. 2823–2824.
[10] D. Shao, T. Zhang, K. Mannar, Y. Han, Time series forecasting on engineering systems using recurrent neural networks, in: Proceedings of the Twelfth International Conference on Advanced Data Mining and Applications, Springer, Gold Coast, QLD, Australia, 2016, pp. 459–471. December 12–15
[11] Y. Kara, M.A. Boyacioglu, Ö.K. Baykan, Predicting direction of stock price index movement using artificial neural networks and support vector machines: the sample of the istanbul stock exchange, Expert Syst. Appl. 38 (5) (2011) 5311–5319.
[12] B. Zhao, H. Lu, S. Chen, J. Liu, D. Wu, Convolutional neural networks for time series classification, J. Syst. Eng. Electron. 28 (1) (2017) 162–169.
[13] B.V. Kini, C.C. Sekhar, Large margin mixture of ar models for time series classification, Appl. Soft Comput. 13 (1) (2013) 361–371.
[14] A. Antonucci, R. De Rosa, A. Giusti, F. Cuzzolin, Robust classification of multivariate time series by imprecise hidden Markov models, Int. J. Approx. Reason. 56 (2015) 249–263.
[15] D. Kulić, W. Takano, Y. Nakamura, Incremental learning, clustering and hierarchy formation of whole body motion patterns using adaptive hidden Markov chains, Int. J. Robot. Res. 27 (7) (2008) 761–784.
[16] R.J. Kate, Using dynamic time warping distances as features for improved time series classification, Data Min. Knowl. Discov. 30 (2) (2016) 283–312.
[17] D.J. Berndt, J. Clifford, Using dynamic time warping to find patterns in time series., in: Proceedings of the KDD Workshop, Seattle, WA, 10, 1994, pp. 359–370.
[18] T.W. Liao, Clustering of time series data survey, Pattern Recognit. 38 (11) (2005) 1857–1874.

[19] X. Zhong, D. Enke, Forecasting daily stock market return using dimensionality reduction, Expert Syst. Appl. 67 (2017) 126–139.
[20] B. Pourbabaee, M.J. Roshtkhari, K. Khorasani, Deep convolution neural networks and learning ECG features for screening paroxysmal atrial fibrillation patients, IEEE Trans. Syst. Man Cybern. Syst. PP (99) (2017) 1–10.
[21] S. Deegalla, H. Bostrom, Reducing high-dimensional data by principal component analysis vs. random projection for nearest neighbor classification, in: Proceedings of the Fifth International Conference on Machine Learning and Applications, IEEE, 2006, pp. 245–250.
[22] B. Seijo-Pardo, I. Porto-Díaz, V. Bolón-Canedo, A. Alonso-Betanzos, Ensemble feature selection: homogeneous and heterogeneous approaches, Knowl. Based Syst. 118 (2017) 124–139.
[23] A. Kayal, A neural networks filtering mechanism for foreign exchange trading signals, in: Proceedings of the IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS), 3, IEEE, 2010, pp. 159–167.
[24] C.S. Vui, G.K. Soon, C.K. On, R. Alfred, P. Anthony, A review of stock market prediction with artificial neural network (ann), in: Proceedings of the IEEE International Conference on Control System, Computing and Engineering (ICCSCE), IEEE, 2013, pp. 477–482.
[25] L. Di Persio, O. Honchar, Artificial neural networks architectures for stock price prediction: comparisons and applications, Int. J. Circuits Syst. Signal Process. 10 (2016) 403–413.
[26] I.P. Marković, M.B. Stojanović, J.Z. Stanković, M.M. Božić, Stock market trend prediction using support vector machines, Facta Univ. Ser. Autom. Control Robot. 13 (3) (2014) 147–158 http://casopisi.junis.ni.ac.rs/index.php/FUAutContRob/article/view/585.
[27] N.I. Sapankevych, R. Sankar, Time series prediction using support vector machines: a survey, IEEE Comput. Intell. Mag. 4 (2) (2009) 24–38.
[28] L.C. Martinez, D.N. da Hora, J.R.d.M. Palotti, W. Meira, G.L. Pappa, From an artificial neural network to a stock market day-trading system: a case study on the BM&F BOVESPA, in: Proceedings of the International Joint Conference on Neural Networks, IEEE, 2009, pp. 2006–2013.
[29] Z. Guo, H. Wang, J. Yang, D.J. Miller, A stock market forecasting model combining two-directional two-dimensional principal component analysis and radial basis function neural network, PLoS ONE 10 (4) (2015) e0122385.
[30] E. Bastı, C. Kuzey, D. Delen, Analyzing initial public offerings' short-term performance using decision trees and SVMs, Decis. Support Syst. 73 (2015) 15–27.
[31] M. Ballings, D. Van den Poel, N. Hespeels, R. Gryp, Evaluating multiple classifiers for stock price direction prediction, Expert Syst. Appl. 42 (20) (2015) 7046–7056.
[32] M. Bildirici, Ö.Ö. Ersin, Improving forecasts of GARCH family models with the artificial neural networks: an application to the daily returns in Istanbul stock exchange, Expert Syst. Appl. 36 (4) (2009) 7355–7362.
[33] M.A. Boyacioglu, D. Avci, An adaptive network-based fuzzy inference system (ANFIS) for the prediction of stock market return: the case of the Istanbul stock exchange, Expert Syst. Appl. 37 (12) (2010) 7908–7912.
[34] A.Ç. Pehlivanlı, B. Aşıkgil, G. Gülay, Indicator selection with committee decision of filter methods for stock market price trend in ISE, Appl. Soft Comput. 49 (2016) 792–800.
[35] A. Gensler, J. Henze, B. Sick, N. Raabe, Deep learning for solar power forecasting an approach using autoencoder and LSTM neural networks, in: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC), IEEE, 2016, pp. 002858–002865.
[36] D.L. Marino, K. Amarasinghe, M. Manic, Building energy load forecasting using deep neural networks, in: Proceedings of the Forty-Second IECON Annual Conference on Industrial Electronics Society, IEEE, 2016, pp. 7046–7051.
[37] A. Grover, A. Kapoor, E. Horvitz, A deep hybrid model for weather forecasting, in: Proceedings of the Twenty-First ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2015, pp. 379–386.
[38] A.C. Türkmen, A.T. Cemgil, An application of deep learning for trade signal prediction in financial markets, in: Proceedings of the Twenty-Third Signal Processing and Communications Applications Conference (SIU), IEEE, 2015, pp. 2521–2524.
[39] G.B. Penn, H. Jiang, O.A.M. Abdelhamid, A.-r. S. A.-r. Mohamed, System and method for applying a convolutional neural network to speech recognition, 2015, US Patent 9190053.
[40] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, L. Fei-Fei, Large-scale video classification with convolutional neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1725–1732.
[41] Y. Kim, Convolutional neural networks for sentence classification, (2014). arXiv:1408.5882.
[42] X. Ding, Y. Zhang, T. Liu, J. Duan, Deep learning for event-driven stock prediction., in: Proceedings of the International Joint Conferences on Artificial Intelligence, 2015, pp. 2327–2333.
[43] D. Kumar, S.S. Meghwani, M. Thakur, Proximal support vector machine based hybrid prediction models for trend forecasting in financial markets, J. Comput. Sci. 17 (2016) 1–13.
[44] D.G. Kleinbaum, M. Klein, Maximum likelihood techniques: an overview, in: Logistic Regression, Springer, 2010, pp. 103–127.
[45] J. Ngiam, Z. Chen, D. Chia, P.W. Koh, Q.V. Le, A.Y. Ng, Tiled convolutional neural networks, in: Proceedings of the Advances in Neural Information Processing Systems, 2010, pp. 1279–1287.
[46] Y. LeCun, K. Kavukcuoglu, C. Farabet, Convolutional networks and applications in vision, in: Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS), IEEE, 2010, pp. 253–256.
[47] N. Amjady, A. Daraeepour, Design of input vector for day-ahead price forecasting of electricity markets, Expert Syst. Appl. 36 (10) (2009) 12281–12294.
[48] M. Längkvist, L. Karlsson, A. Loutfi, A review of unsupervised feature learning and deep learning for time-series modeling, Pattern Recognit. Lett. 42 (2014) 11–24.
[49] Y. Bengio, A. Courville, P. Vincent, Representation learning: a review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (8) (2013) 1798–1828.
[50] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, F.E. Alsaadi, A survey of deep neural network architectures and their applications, Neurocomputing 234 (2017) 11–26.
[51] Y. Zheng, Q. Liu, E. Chen, Y. Ge, J.L. Zhao, Time series classification using multi-channels deep convolutional neural networks, in: Proceedings of the International Conference on Web-Age Information Management, Springer, 2014, pp. 298–310.
[52] H. Gunduz, Z. Cataltepe, Y. Yaslan, Stock market direction prediction using deep neural networks, in: Proceedings of the Twenty-Fifth Signal Processing and Communications Applications Conference (SIU), IEEE, 2017, pp. 1–4.
[53] A. Özgür, L. Özgür, T. Güngör, Text categorization with class-based and corpus-based keyword selection, in: Proceedings of the International Symposium on Computer and Information Sciences, Springer, 2005, pp. 606–615.
[54] M. Steinbach, G. Karypis, V. Kumar, et al., A comparison of document clustering techniques, in: Proceedings of the KDD workshop on text mining, Boston, 400, 2000, pp. 525–526.
[55] F. Chollet, et al., Keras, 2015, (https://github.com/fchollet/keras).
[56] N. Srivastava, G.E. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting., J. Mach. Learn. Res. 15 (1) (2014) 1929–1958.
[57] R. Caruana, S. Lawrence, C.L. Giles, Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping, in: Proceedings of the Advances in Neural Information Processing Systems, 2001, pp. 402–408.
[58] S. Barak, M. Modarres, Developing an approach to evaluate stocks by forecasting effective features with data mining methods, Expert Syst. Appl. 42 (3) (2015) 1325–1339.
[59] X. Zhong, D. Enke, A comprehensive cluster and classification mining procedure for daily stock market return forecasting, Neurocomputing 267 (2017) 152–168.
[60] R.C. Cavalcante, R.C. Brasileiro, V.L. Souza, J.P. Nobrega, A.L. Oliveira, Computational intelligence and financial markets: a survey and future directions, Expert Syst. Appl. 55 (2016) 194–211.
[61] A. Özgür, T. Güngör, Classification of skewed and homogenous document corpora with class-based and corpus-based keywords, in: Proceedings of the Annual Conference on Artificial Intelligence, Springer, 2006, pp. 91–101.
[62] H. Song, A.K. Qin, F.D. Salim, Multivariate electricity consumption prediction with extreme learning machine, in: Proceedings of the International Joint Conference on Neural Networks (IJCNN), IEEE, 2016, pp. 2313–2320.
[63] J. Wan, J. Liu, G. Ren, Y. Guo, D. Yu, Q. Hu, Day-ahead prediction of wind speed with deep feature learning, Int. J. Pattern Recognit. Artif. Intell. 30 (05) (2016) 1650011.