International Journal of Applied Science and Engineering
2018. 15, 3: 183-198

Deep Learning Applications in Pattern Analysis Based on Polar Graph and Contour Mapping

Yi-Chi Tsai[a] and Tsung-Nan Chou[b*]

[a]Department of Computer Science and Information Engineering, Chaoyang University of Technology, Taichung, 413, Taiwan, R.O.C.
[b]Department of Finance, Chaoyang University of Technology, Taichung, 413, Taiwan, R.O.C.

Abstract: In recent years, small businesses and companies across industries have confronted the challenges of big data analysis since a huge amount of time series and cross-sectional data are collected daily from their business activities. These data might require dimensionality reduction to reduce the complexity of data analysis. This study used two special feature patterns including polar and contour graphs to represent the mutual relationship of the Asian stock markets, and geometric and invariant measures were applied to replace the original variables with an attempt to reduce the dimensionality of input variables.

Moreover, several feature selection and extraction methods were also implemented with predictive models in comparison to the effectiveness of proposed feature patterns. Three conventional machine-learning approaches were employed as benchmark models to predict the daily change of Taiwan stock market index based on the transformed graphs. In addition, the predictive performance of another three deep learning approaches including stacked autoencoder, deep neural network, and convolution neural network were evaluated with different network structures and feature selection strategies. Based on organizing the input variables as multivariate time series and two-dimensional encoded images, the experiment results showed that the deep learning models outperformed other machine learning models. Especially, the predictive accuracy of the CNN model with 1D convolution improved the predictive accuracy to 0.67 with a precision of 0.51 and a recall of 0.55, respectively. The results also suggested that the transformation of two-dimensional polar and contour graphs might provide abstract features for the deep learning approaches to explore the historical patterns. As the CNN models with 1D and 2D convolution performed better than the rest of deep learning models, the further studies might be needed to investigate how both models complement each other to create synergies.

Keywords: Deep learning; dimensionality reduction; polar graph; contour mapping.

1. Introduction

Since the business world has been increasingly driven by new technologies such as big data analysis, artificial intelligence and data mining recently, a lot of small and medium business companies struggle with how to select the efficient information techniques suitable for their data strategy and try to explore massive data for values. Therefore, how the business companies choose the right analytical techniques to extract insights and information from data become significantly importance. Because of the fast improvement of machine learning and related data automation

[*]Corresponding author; e-mail: tnchou@cyut.edu.tw

doi: 10.6703/IJASE.201812_15(3).183

Received 1 November 2017
Revised 4 July 2018
Accepted 5 July 2018

techniques, a lot of companies encounter great difficulties in selecting an ideal solution to process and analyze their data with efficiency. In the case of financial data such as credit card transactions and stock screening data, the demands for employees with the skill of big data analysis are crucial to the future success of financial institutions. However, most companies have limited knowledge of processing and analyzing high-dimensional or diversified data, resulting in poor and incorrect analysis results for their corporate decision-making problems. On the other hand, some companies only focus on how to manage the huge amounts of data and related equipment needed for business operations, but ignore the data integrity and the impact of their update speed to the analytical models. As a result, the companies will find inability to make more accurate and effective decisions based on lengthy data analysis processes. Meanwhile, the data source varies from organization to organization and might include missing, redundant, biased, repetitious values. Accordingly, the collected data require complex data preprocessing and even deal with extreme or outlier for any variable. The increasing dimensionality of data also create another problem with the data analysis and deteriorate the analytical model to produce satisfied results. A suitable dimensionality reduction for data analysis will help to improve the accuracy of any data mining models.

The transformation from raw input data to the feature space could be different according to different domains knowledge for industries. In the field of computer vision and pattern recognition, the location, geometry, color, and texture of the sensed targets within an image often are referred as input features for classification and identification. Unlike the image data, the data collected for a business organization normally consist of panel data with numerical and categorical variables. These data include time series and cross-section characteristics that compose with high dimensionality and change dynamically over time. Therefore, if the input data of a predictive model are directly collected from the enterprise without any transformation, the performance of the model is limited because the raw data contains redundant information, and multiple variables in a large data set often generate complex feature spaces with mutual interference. In order to reduce the computational cost of predictive models and increase their efficiency, the collected data require dimensionality reduction to transform all features from high-dimensional data space into a lower dimension. In general, the data reduction methods comprises feature extraction and feature selection, but the latter is normally preferable if the users intend to remove the irrelevant variables. The feature extraction projects data group from a high-dimensional space into a low-dimensional space by means of a linear or nonlinear combination of variables to create new features or variables, whereas the feature selection chooses partial features from the original data space and expects to achieve better prediction or classification results. Unlike the above two reduction methods, the feature engineering focuses on generating new customized features based on various encoding methods through continual observations.

A great deal of research work conducted on the various techniques to reduce the data dimensionality was introduced and summarized in [1]-[3]. The most common data reduction approaches were the use of statistical methods such as principal component analysis (PCA) and linear discriminant analysis [4, 5]. Besides, some research works applied other methods such as rough sets theory (RST), decision tree (DT) and gray relational analysis (GRA) to screen out unimportant features [6, 7]. In this paper, two special feature patterns including polar and contour graphs were designed to represent the mutual relationship of the Asian regional stock markets, and their geometric and invariant metrics were used to replace the original variables with an attempt to reduce the dimensionality of data. Meanwhile, other feature selection and extraction methods were also evaluated with predictive models in comparison to our proposed feature patterns. Several machine learning (ML) and deep learning (DL) approaches were employed to predict the daily change of Taiwan stock market index, and their performances were evaluated and summarized in this study. The remainder of the paper was organized as follows. In Section 2, we introduced research methods and experiment design. Section 3 described the experimental results, and finally, we addressed conclusion and future work in Section 4.

2. Research Methods and Experiment Design

For this study, the financial market data collected over the course of seven years were used in experiments to explore the performance of predictive models and algorithms. The input data comprised the daily changes of 22 Asian stock markets and subdivided into 1500 for the training dataset and another 250 for the testing dataset. The aggregation of these data was referred to as original variables thereafter in this study and used by the predictive models to train and forecast the movement of a price up or down next day. To deal with the data differed with units and scale, both the training and testing datasets were normalized to a common scale prior to the training process. In order to predict the trend direction of a stock market, the daily change of Taiwan stock market index was divided into ten intervals. Unlike the intraday change, the daily change calculated the difference between a given day's close of a stock index and the close of its previous trading day. Since the gain or loss of a trading day was converted to ten intervals according to the distribution of training samples, the middle intervals might span over both positive and negative values in terms of price movement. Therefore, these intervals were regarded as highly variation and might be difficult for the models to predict the direction of a market index. The experiment designs incorporated two special feature patterns to represent the mutual relationship of the Asian regional stock markets, and totally eleven experiments categorized into two groups, machine learning (ML) and deep learning (DL), were set up to predict the daily change of Taiwan stock market index.

The first experiment used the original variables as the initial inputs to three machine-learning approaches. As three ML models produced incorrect and even contradictory outcomes, the evidential reasoning method based on Dempster Shafer theory [8, 9] was involved in the second experiment to examine whether the combination of three ML models improved the predictive performance. Since high-dimensional data increased the analytical difficulty for any models, some researchers [10] suggested various solutions based on feature extraction and selection to remove the irrelevant variables for data analysis. Therefore, to reduce the dimensionality of input data, the top seven and nine principal components derived from the principal component analysis (PCA) were chosen to unite with ML models in the third experiment in comparison with the moment approaches using the same number of variables. Because the original input variables were transformed to a two-dimensional polar graph, another two Zernike’s moments were appended with seven Hu’s moments in the fourth experiment to represent the features of a polar graph for reducing its dimensionality. Several shape descriptors were also used to measure the 2D image and examined with the ML models in the fifth experiments.

In addition to the conventional ML models, the rest of experiments employed the newly developed deep learning approaches with various configurations. Three deep learning algorithms including stacked autoencoder (SAE), deep neural network (DNN) and convolution neural network (CNN) were evaluated with different network structure and feature selection. Some experiments direct imported the polar and contour images as inputs, so the number of input variables was increased to 784 variables if the image was converted to one-dimensional vectors. The SAE model was evaluated for such high dimensional inputs through encoder and decoder conversion. On the other hand, the DNN and CNN models were evaluated with different feature selection strategies. Three types of feature selection algorithm including correlation analysis, information gain and decision tree were incorporated to filter the original variables and aimed to reduce the feature dimension with little losing accuracy. Applying feature selection and extraction could reduce the complexity of a forecasting model, but the prediction accuracy might decrease due to losing the data integrity. On the other hand, using feature engineering to explore a new set of creative variables might increase the number of variables and complexity of computation for analytical models, but the fast development of artificial intelligence (AI) hardware and deep learning techniques enabled us to explore and train the high-dimensional data with efficiency. Because the input data was 2D image and the number of variables was expanded to 784, the rest of experiments examined how the deep learning neural network dealt with such a high-dimensional time series data with improved accuracy and efficiency than machine learning approaches. The following sections introduced the methods and strategies used in this study briefly.

2.1. Polar Graph and Contour Mapping

As illustrated in Figure 1, the original 22 input variables were converted into a two-dimensional polar graph according to their normalized values, which were expressed in polar coordinates with different radius and angle. Moreover, the polar graph was encoded as a binary image with 28-by-28 pixels for the calculation of invariant moments and shape measures. The polar graph not only illustrated the complexity of input variables with picture but also could be transformed with flexibility by measuring its properties to reduce the number of variables.

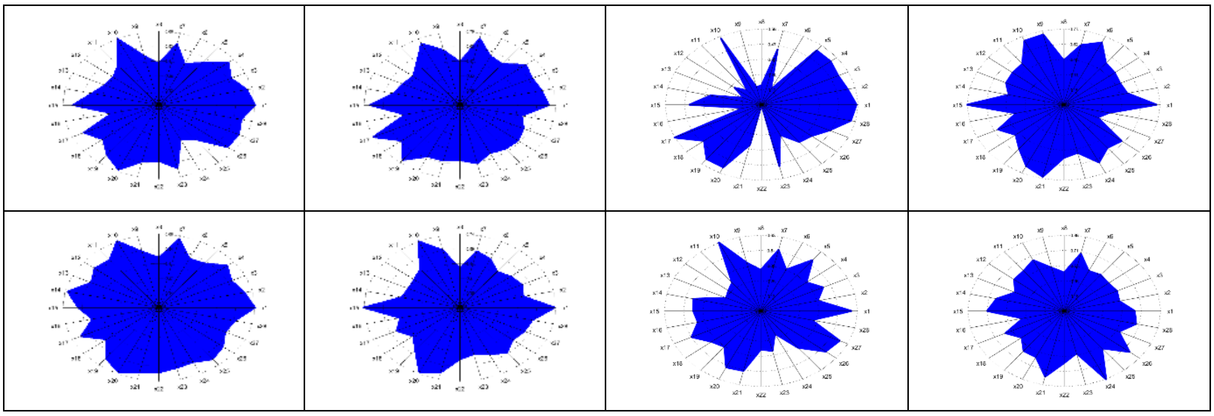

Figure 1. Examples of the 2D polar graph.

The contour graph was another pattern transformation in our experiments. The original 22 variables were combined with their lagged time series data based on various length of time intervals, and both the spatial and temporal data were reorganized into a three-dimensional formation. The formation represented the 3D geometric map of input data within a specific time frame and required further 2D transformation for the deep learning approaches to perform pattern recognition. The 3D geometric map could be easily scaled to different time frequency such as weekly, monthly, quarterly and annual interval to learn the historical trend of a stock market index. The Figure 2 shows the three-dimensional geometric map based on a monthly interval and its initial transformed 2D contour graph.

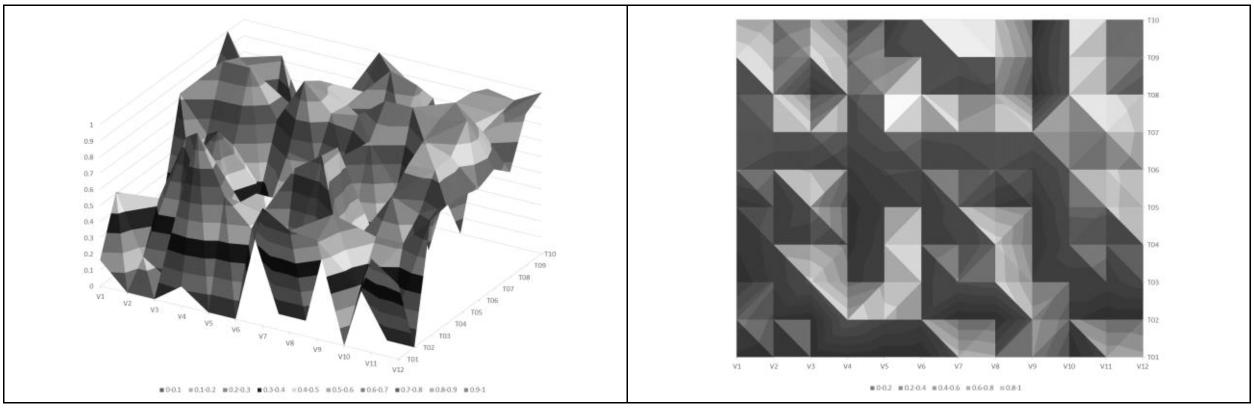

Figure 2. Example of 3D geometric map vs. initial 2D contour graph.

To enable the predictive model to learn the features embedded in the 3D geometric map, the 3D map was sliced and decomposed into a two-dimensional contour graph layer by layer according to the specified threshold value. The contour mapping could produce any number of layers as needed in this study to examine the potential insight of data. Figure 3 below displays four sliced layers from a 3D geometric map.

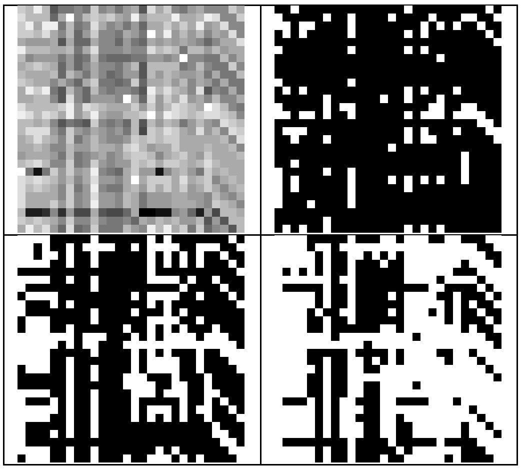

Figure 3. Sliced 2D contour graphs.

2.2. Shape and Invariance Measures

As stated in the previous section, all 22 variables including numerical and categorical data were transformed into a 2D image of polar and contour graph, the training and testing of models were directly performed on these images. In practice, these images were generally analyzed and classified by their shape descriptors. The basic feature extraction applied in this study used a set of shape descriptors to measure the 2D polar graph. The collected descriptors included nine metrics. The Area and Centroid measured the number of pixels within the polar graph and its corresponding central coordinates. On the other hand, Axis Length and Eccentricity measured the surrounding ellipse of the polar graph. The Equivalent Diameter searched for the diameter of a circle that gave the same area as the polar graph. In addition, both the Solidity and Extent measured the ratio of the area between the polar graph and its bounding rectangle and convex hull respectively. Finally, the Bounding Box calculated the smallest rectangle cover the polar graph, and its top-most and right-most coordinates from the center provided another two optional metrics. However, to measure the 2D shapes differed with rotation, translation, and scale within an image was a matter of considerable difficulty. Therefore, the Zernike moments [11] and Hu moments [12] were adopted in this study as other metrics because of their invariant properties.

The images processed by Hu moments created geometric features that were invariant to rotational, scale and translational transformations. In other words, the 2D shapes within an image could be measured by different order moments. In that case, high order moments tended to capture a lot of noise and result in imprecise outcomes. By using Hu's moments, the centroid of target shape was calculated by taking the quotient of the first order moment over the zeroth-order moment, and then any other moments computed with this centroid were referred as central moments that indicated the deviations from the centroid. The first order central moment provided the average pixel value for target shape and represented the mean of the specified shape. The second order central moment was regarded as the standard deviation that was the square root of the variance of the distribution. The third order central moment represented skewness that gave a measure of the degree of asymmetry. If a target shape had a large negative third order moment, the target shape was skewed to the left. The fourth order central moment represented kurtosis and it represented the measure of peakedness in the distribution. Totally seven Hu's invariant moments were used in this study to classify a target shape.

Unlike geometric moments, Zernike moments calculated independently to arbitrary high orders without having to recalculate low order invariants. The Zernike moments including phase (angle) and amplitude of the moment were calculated on a binary image according to a mentioned order and repetition number. Using Zernike moments with different order and repetition, the necessary features were determined and extracted for classification. Further, by preprocessing an image using both the Hu and Zernike moments, we could capture more shape and boundary information differed from the shape descriptors.

2.3. Dimensionality Reduction Approaches

In addition to use the invariance measures to reduce the dimensionality of data for the polar graph, other feature selection and extraction approaches based on Pearson correlation analysis, information gain, decision tree, and principal component analysis were also applied in this study. Despite the fact the bivariate correlations method troubled with the problem of multicollinearity to select relevant variables, the Pearson correlation analysis still could be used to explore the strength and direction of linear relationships between the variables based on the assumption that the variables were normally distributed. The correlation test in experiment chose the top seven variables, which were highly relevant to the dependent variable, as inputs for models.

Additionally, information gain was another alternative for feature selection in this study. The entropy of each input variable was calculated for the dependent variable, and the input variables that contribute more information were selected. The decision tree was the third alternative of feature selection employed to discover the importance of the input variables based on evaluating the model performance with different subsets of data. The resulting antecedents of decision rules were used to choose important valuables. In addition, the principal component analysis was another method been applied for dimensionality reduction with ML models in the third experiment. However, this method only used for providing the initial comparison with the outcomes of the shape and invariance measures, because it was a statistical method commonly used to evaluate a set of raw features for dimensionality reduction.

2.4. Applied Deep Learning and Other Machine Learning Approaches

Since different strategies of dimensionality reduction were evaluated with three machine-learning approaches in the preliminary experiments, their performances were used as benchmark and compared with the newly developed deep learning approaches. The machine-learning techniques (referred as ML models thereafter) including naive Bayes (NB), decision tree (DT), and artificial neural network (ANN) addressed in [13] were employed to predict the daily change of stock market index by using the original variables. The NB method was a supervised learning algorithm based on Bayes theorem and worked very well in many research applications. This approach only required a small amount of training data to estimate the required metrics. On the other hand, the C4.5 decision tree, improved from the conventional ID3 algorithm, could handle training data with missing values and performed classification tasks with improved computational efficiency. Meanwhile, the reason to use a neural network as the benchmark model was that its performance could be compared to the deep learning algorithms later, which was the latest development of the neural network. The hidden layer of a neural network was defined as the average number of input variables and output variables in our experiment. For speeding up the convergence of the training process, the learning rate and momentum were set to 0.5 and 0.2 respectively. The input variables were also normalized prior to the training process of the neural network.

The most common type of the deep learning approaches (referred as DL models thereafter) [14, 15] included the deep neural network (DNN), developed from the conventional multilayer perceptron (MLP), but was designed with more complex architectures. The DNN model, when compared to the artificial neural network (ANN), involved advanced features such as more hidden layers including dropout layers to prevent overfitting, and each layer consisted of neurons with more sophisticated activation functions. The input and output layer could be tailored and customized with high-dimensional input patterns against target variables. Many other algorithms and techniques for deep learning models were pervasively developed recently including the most favorable approaches such as the stacked autoencoder (SAE), convolutional neural network (CNN) and recurrent neural network (RNN). The MLPs worked well on transactional tabular data, however, if the image data was considered as inputs, then CNN model [16, 17] might provide a better solution to discover the hidden features of the image. In contrast, if the sequential data such as audio signal and time series data was imported as inputs, then RNN model might be a good choice for these data.

An autoencoder implemented with neural networks was an unsupervised learning algorithm that performed self-learning with the compression (encode) and decompression (decode) functions to explore the abstract representations of data. Since the autoencoder was a data-specific approach and intended to learn the representation of features without the need to specify their corresponding labels in advance, its performances only depended on what it was trained. This study used a stacked autoencoder neural network to discover the abstract structure of input data with different sparse representations and enabled both historical polar and contour graphs to be trained for prediction tasks. As the input layer of SAE neural network was the 784-pixel intensity values converted from a 28×28 image, the SAE was designed to incorporate a different number of neurons in hidden layers to learn the encoded representation of the polar and contour graphs. With appropriate dimensionality and sparsity constraints, SAE could learn data projections that were more efficient than PCA or other techniques. As illustrated in Figure 4, two autoencoders, each with multiple layers, were concatenated and the outputs of the first autoencoder was the inputs of the successive autoencoder. In addition, the final outputs were decided by another Softmax layer. Unlike the sigmoid function used as activation function in layers of a conventional neural network or logistic regression model in binary classification, the deep learning model applied Softmax function to calculate the probabilities of each target class over all possible classes for multi-class classification problems. The output probabilities ranged from 0 to 1 for each class were used to predict the target class, and the sum of all the probabilities was equal to one.

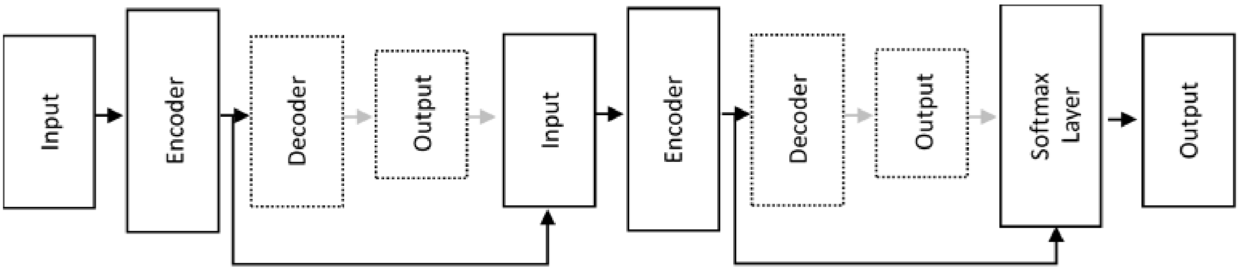

Figure 4. Stacked Autoencoder neural network.

The architecture of neural networks applied at this stage was considered feedforward rather than recurrent because we were dealing with the transformed 2D image as inputs. In comparison to the conventional ANN model required only single or fewer hidden layers in structure, the DNN models in this study were composed of more hidden layers, and each layer performed with different sophisticated functions. For example, the hidden layer was fulfilled with different kernels and activation functions to extract features. Meanwhile, a dropout layer with different dropout rates was inserted between hidden layers as a regularization to prevent from overfitting of models. Various activation functions such as Relu, Sigmoid and Softmax were tested to find the optimal results for each hidden layer. However, increasing network depth and involving complex activation functions might degrade the network performance during the training process. Therefore, three different network structures constructed with three, four and five hidden layers were used in experiments and denoted as DNN-S1, DNN-S2, and DNN-S3 respectively. The number of neurons in each layer was decreased from 128 to 16 depending on how many layers implemented. On the other hand, the original 22 variables were combined with their corresponding time series data based on specific length of time, and both the spatial and temporal data were arranged in a two-dimensional formation, which represented as a feature pattern for further training by the CNN models. The architecture of CNN models applied in this study included six different layers. The feature pattern consisting of 28x28 input variables was used as inputs to the first layer of the network. The second layer was the first hidden layer of the network and performed basic convolution operations over the input pattern. A 5x5 kernel was used to filter the input pattern, and totally six distinctive kernels were employed as filter mask in a convolutional layer to exploit the abstract features of patterns. The third layer was a max pooling layer, where a small 2x2 pooling region was sampled to reduce the output of the previous layer. In practical, the maximum value of that region was taken as the output of this layer and the dimensionality of input pattern was reduced by eliminating non-maximal values. The fourth layer was another convolution layer applying the same 5x5 kernel to filter the output of the previous layer through another 12 distinctive kernels. The fifth layer was a max pooling layer again which identical with the previous one in the third layer. Finally, the last layer was fully connected and correspond to a traditional neural network. The softmax function employed in this layer classified the outcomes into ten different intervals, and an associated dynamic threshold mechanism was designed to interpret the intervals to the final categories of stock trend direction. To explore the prediction performance of various convolutional neural networks, another network structure composed of 10 and 20 distinctive kernels for each convolutional layer was also evaluated.

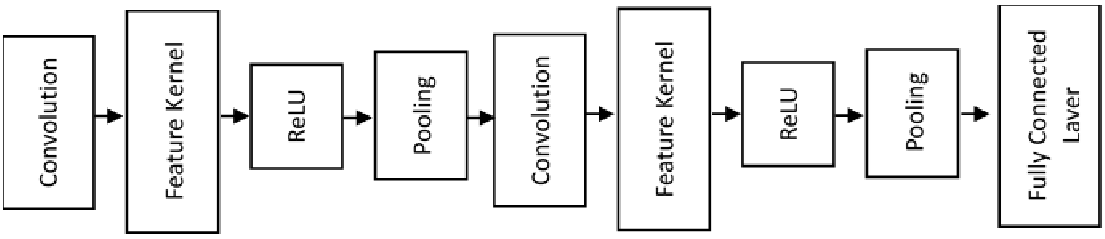

Figure 5. Convolution neural network for 1D and 2D convolution.

The Figure 5 illustrated a basic convolution neural network with multiple hidden layers which was used to solve various image classification and recognition problems under complex scenarios. Each hidden layer learned the abstraction from the output of previous layer. However, training neural networks with multiple hidden layers was time-consuming if the structure of network became deeper with various layers and functions.

Since the collected data included time series data, and the CNN model with 1D convolution was faster than 2D convolution. Therefore, both 1D and 2D models were employed in this study to compare their performance. The 1D convolution sliced with a window composed of predefined kernels across the data with padding. The structure of CNN models was modified to include seven different layers for 1D convolution. The input layer used top ranked 8 and 16 variables from Pearson correlation analysis and followed by two 1D convolution layers with 32 filters, the filter for each convolution layer was a one-dimensional kernel of five and three respectively. Another dropout layer, max pooling layer and a final layer with sigmoid activation function were also appended to the CNN model with 1D convolution. The 1D convolutions with 8 or 16 variables were denoted as CNN1D-8 and CNN1D-16 models.

3. Results and Discussion

The experiments conducted in this section comprised two stages. The first stage involved five experiments which applied three conventional machine-learning algorithms with feature extraction and selection to differentiate polar graph. The second stage integrated the deep learning technique with various algorithms and comprised another three experiments. As the confusion matrix contained information about actual and predicted outcomes, several metrics including accuracy, sensitivity, specificity, precision, and F-score were employed by the models to compare with each other. The results of experiments were described as follows.

3.1 Experiment Results of Conventional Machine-Learning Approaches

The first experiment directly used the original 22 variables as inputs, and three machine-learning algorithms were evaluated for their predictive accuracy. As shown in Table 1, among all three models, the ANN performed better than the other two models. However, although the specificity reached higher than 0.67 for all models, their predictive accuracy was less than 0.5 and considered worse than the below-chance accuracy.

Table 1. Results for ML models with original 22 variables.

ML Models (22 Var.)	Accuracy	Sensitivity	Specificity
NB	0.45	0.37	0.68
DT	0.43	0.36	0.68
ANN	0.46	0.34	0.67

As all three models produced a collection of predictive outcomes, which comprised incorrect, ambiguity or even contradictory results against each other, the DST reasoning method was then incorporated in the second experiment to synthesize the outcomes of models with the aim of reducing the overall predictive errors. The fusing results for any two ML models were listed in Table 2. The combination of the NB and ANN models acquired the higher predictive accuracy although it still regarded as slightly better than the below-chance accuracy. Alternatively, all the outcomes of three ML models were fused again by the DST algorithm, but the performance only revealed a minor improvement and achieved an accuracy of 0.53.

Table 2. Results for ML models with DST Fusion.

DST Fusion	Accuracy	Sensitivity	Specificity
NB+DT	0.51	0.38	0.70
NB+ANN	0.52	0.38	0.70
ANN+DT	0.49	0.35	0.68
NB+NN+DT	0.53	0.39	0.71

To demonstrate the performance of statistical feature extraction, the principal component analysis was examined in the third experiment to reduce the dimensionality of input data. The new inputs, which were the linear combinations of the original 22 variables, comprised the top nine and seven principal components derived from PCA method respectively. As demonstrated in Table 3, all of the performance metrics obtained poor results similar to the previous experiments. Only the NB model reached a slightly higher accuracy at 0.51, and the result was not regarded as better than the accuracy of random guessing again.

Table 3. Results for ML models with PCA.

Principal Components	Models	Accuracy	Sensitivity	Specificity
(Top 9 Var.)	NB	0.49	0.36	0.69
	DT	0.44	0.32	0.66
	ANN	0.50	0.36	0.69
(Top 7 Var.)	NB	0.51	0.37	0.70
	DT	0.45	0.33	0.67
	ANN	0.49	0.36	0.69

Since the original 22 variables were transformed to a two-dimensional polar pattern, we used the invariant moments to describe the polar pattern for the reason of reducing the dimensionality of data. Although seven Hu's moments calculated from central moment were applied in the fourth experiment to represent the rotation, translation, and scale invariant measures of patterns, another two Zernike's moments were appended later to evaluate whether more invariants could improve the prediction accuracy. By referring to Table 4, it was obvious that the performance of all models failed to outperform the random guessing accuracy.

Table 4. Results for ML models with moments.

Invariant Moments	Models	Accuracy	Sensitivity	Specificity
Hu & Zenkit Moments	NB	0.49	0.36	0.69
	DT	0.47	0.34	0.67
	ANN	0.48	0.35	0.68
Hu Moments	NB	0.50	0.36	0.69
	DT	0.46	0.33	0.67
	ANN	0.45	0.32	0.66

In addition to the invariant moments, the shape measures, including the Area, Centroid, Axis Length, Eccentricity, Equivalent Diameter, Solidity, Extent, and Bounding Box were also used to describe the 2D polar graph and worked as inputs to ML models. Table 5 below indicated all performance metrics obtaining poor results the same as the previous experiments.

Table 5. Results for ML models with shape measures.

Measures	Models	Accuracy	Precision	F-score
Shape and Bounding Box	NB	0.51	0.46	0.48
	DT	0.46	0.22	0.30
	ANN	0.49	0.48	0.40
Shape only	NB	0.51	0.46	0.51
	DT	0.46	0.22	0.29
	ANN	0.51	0.50	0.45

3.2 Experiment Results of Deep Learning Approaches

Despite having unsatisfactory results in the previous sections, we then employed both the 2D polar graph and contour graph as input images to the deep learning models. The contour graph reconstructed the original 22 variables into the formation of a three-dimensional geometric map to illustrate the spatial and temporal relationship of variables. The 3D geometric map was later sliced with a specified threshold value to create 2D images for analysis. As a result, the number of input variables was largely increased to 784 if the two-dimensional matrix of an image was converted to a one-dimensional vector. Both the polar graph and the sliced contour graph were evaluated with the SAE model of deep learning approaches. In Table 6, the result indicated that the SAE model performed better with the polar graph, and achieved an accuracy of 0.56, which outperformed all ML models.

Table 6. Result for DL models with stacked autoencoder.

SAE Model	Accuracy	Sensitivity	Specificity
Polar Graph	0.56	0.57	0.54
Contour Mapping	0.52	0.75	0.25

In comparison to the simpler SAE model, another DNN model built with more complicated structure was evaluated in the next experiment. The DNNs were implemented with three, four and five hidden layers respectively, and each layer comprising the number of neurons from 128 to 8. The experiment results in Table 7 denoted that the best result was the DNN-S3 model with the accuracy of 0.56, which outperforms other models having more hidden layers. However, this result implicated that the number of neurons in each layer was not sensible, and the increase of the hidden layers failed to improve the predictive accuracy substantially.

Table 7. DNN models with different network structure.

DNN (22 var.)	Accuracy	Precision	Recall	F-score
DNN-S1	0.54	0.56	0.46	0.50
DNN-S2	0.53	0.53	0.56	0.55
DNN-S3	0.56	0.56	0.65	0.60

To examine whether dimensionality reduction increased the predictive accuracy for deep learning models, three types of feature selection were integrated with the DNN models, including correlation analysis, information gain analysis and decision tree. The following experiments were divided into three groups, and only the top seven ranked variables from feature selection were selected as inputs to enable provably consistent configuration with the former ML models.

By referring to Table 8, the average accuracy of DNN models with Pearson correlation remained at around 0.56, and the accuracy of model DNN-S3 was slightly lower than the other two models. On the other hand, the DNN models using information gain reported the average accuracy of 0.58 in Table 9. In particular, the DNN-S2 model improved the accuracy to 0.60 and was considered as the best model if compared to the other two groups in feature selection. Table 10 illustrated the experiment result for the feature selection using the decision tree, and all three models fell into the same accuracy of 0.58, but slightly differed in other performance metrics.

Table 8. DNN models with Pearson correlation.

DNN (7 var.)	Accuracy	Precision	Recall	F-score
DNN-S1	0.56	0.56	0.56	0.56
DNN-S2	0.56	0.55	0.65	0.59
DNN-S3	0.55	0.55	0.62	0.58

Table 9. DNN models with Information Gain.

DNN (7 var.)	**Accuracy**	**Precision**	**Recall**	**F-score**
DNN-S1	0.56	0.59	0.47	0.52
DNN-S2	0.60	0.61	0.61	0.61
DNN-S3	0.59	0.59	0.65	0.62

Table 10. DNN models with Decision tree.

DNN Model (7 var.)	**Accuracy**	**Precision**	**Recall**	**F-score**
DNN-S1	0.58	0.59	0.61	0.60
DNN-S2	0.58	0.58	0.63	0.60
DNN-S3	0.58	0.57	0.69	0.62

As a highlight, considering the DNN models with feature selection, all models, excluding the DNN-S3 in the group of Pearson correlation, claimed the accuracy greater than 0.58, and outperform the ML models. Particularly, the DNN-S2 in the group of information gain achieved the highest accuracy, although its F-score was slightly lower than the DNN-S3 in the same group. Normally, it was not easy to choose the right metrics for the specific classification tasks such as identifying the defaulters for the banking industry, which involved an imbalanced classification problem. In this study, if the stock traders considered the upturn experience was their major concern, then the accuracy might not be the only adequate metric, and a higher recall rate (sensitivity) was required for their evaluation result. Consequently, the DNN-S3 model might be the favorable model, even by losing a bit of accuracy.

The final experiment in this study implemented the CNN neural network with both 1D and 2D convolution. As each of the input variables was time series data, the properties hidden in data could be explored and filtered by the specific 1D and 2D kernels during the learning process. Since we transformed the input variables into a two-dimensional 28x28 image and used 2D convolution for the CNN models in our previous study [18], the former experiment results were used here to compare the performance of the CNN model with 1D convolution to prove whether the time series data performed better with 1D or 2D convolution. Two CNN models based on 1D convolution were evaluated with 8 and 16 input variables respectively, and all models implemented with various kernels described in section 2.1. As shown in Table 11, both models performed well and improve the accuracy to the highest of 0.67. The result also outperformed the CNN model with 2D convolution in our previous work.

Accordingly, the CNN-1D-16 model was considered as the most competent model among all experiments in this paper. In addition, the experiment result also indicated more input variables for the CNN1D model might slightly increase its accuracy. Despite the fact that the CNN2D model achieved an accuracy of 0.65, which was superior to all ML models in previous experiments, the accuracy was still worse than the CNN1D models. The experiment results in Table 11 below suggested that time series data could be implemented with CNN models through both the 1D and 2D convolution to acquire moderate results.

Table 11. CNN models with 1D and 2D convolution.

CNN Model	Accuracy	Precision	Recall	F-score
CNN1D-8	0.66	0.46	0.68	0.55
CNN1D-16	0.67	0.51	0.55	0.53
CNN2D-28x28	0.65	0.53	0.52	0.53

4. Conclusion

In this study, the input variables were converted to a two-dimensional polar graph for reducing the feature dimensions based on the various measurements of the graph. Moreover, the input variables were rearranged as the combination of temporal and cross-sectional data, and were transformed to another two-dimensional contour graph derived from its corresponding three-dimensional geometric map. The proposed polar graph was measured by its geometric shape descriptors and invariant moments, which worked as inputs for three conventional machine-learning approaches to evaluate predictive performance. In addition, principal component analysis and evidential reasoning based on the Dempster-Shafer theory were also employed for the feature selection and the fusion of model outcomes respectively. As the proposed strategies above failed to achieve better performance than random guessing accuracy for three machine-learning approaches, the newly developed deep learning approaches including SAE, DNN, and CNN models were applied to examine whether the new techniques could improve the predictive accuracy.

The SAE models based on unsupervised learning was used as the first alternative to evaluate accuracy for both polar and contour graphs. The experiment results suggested that the SAE model outperformed the machine learning models, and the predictive accuracy was improved to 0.56 with a Sensitivity of 0.57 and a Specificity of 0.54 for using the polar graph. Furthermore, the DNN models were evaluated with a different number of hidden layers and neurons, which involved the dropout and pooling layers to improve the training process. Additionally, three feature selection methods including correlation analysis, information gain and decision tree were implemented with the DNN models. The experiment result indicated the DNN models with different network structure increased the accuracy to 0.56. On the other hand, the DNN-S2 model within the group of information gain achieved the highest accuracy of 0.60 and was considered as the best model if compared to the other two groups of feature selection. Finally, the CNN model based on 2D convolution was applied to explore the embedded historical pattern of a sliced contour image, which was constructed by rearranging the original 22 variables and their corresponding time series data into a two-dimensional formation. The resulting 2D image might enable the CNN model using 2D convolution to recognize its historical pattern. The experiment result indicated that the CNN2D model improved the accuracy to 0.65 from the 0.60 if compared to the DNN models.

Although the transformed 2D image, which represented the spatial and temporal correlation of financial time series data, increased the number of variables from 22 to 784, the experiment result demonstrated the encoded 2D image might enable the SAE and CNN2D models to explore the embedded historical trend for a specified time interval. Instead of encoding time series data into a 2D image for pattern recognition by the CNN2D model, the CNN models were also tailored to classify multivariate time-series data without the need to specify the trend patterns manually by using the one-dimensional convolution with various kernels. Comparing to all deep learning models evaluated in previous experiments, the CNN1D model improved the predictive performance from around 0.55 to almost the highest accuracy of 0.67.

In conclusion, the experiment results of this study suggested that the deep learning models outperformed the machine-learning models. Especially, the predictive accuracy of the CNN model with 1D convolution improved the predictive accuracy to 0.67 with a precision of 0.51 and a recall of 0.55, respectively. Although the conventional machine-learning models encountered difficulties in learning high-dimensional input data, our results recommend the performance could be improved by using deep learning algorithms with the applicable arrangement of input variables and network structure. As the dimensionality of original input variables was expanded to a higher dimension due to the 2D graph transformation, the experiment results showed the encoded 2D image might provide a meaningful pattern for the deep learning models to explore the abstract features of the historical stock trend. Finally, it was worth to point out that time series data implemented with CNN models with 1D and 2D convolution achieved the better performances, and the further studies might need to explore how both models complemented each other to create synergies.

Acknowledgements

This work was supported in part by grants from the Ministry of Science and Technology, Taiwan (R.O.C.) under contract number: MOST 105-2633-E-324-001, and a preliminary version of this paper appears in Proceedings of the 2017 IEEE 8th International Conference on Awareness Science and Technology (iCAST 2017).

References

[1] Khalid, S., Khalil, T., and Nasreen S. 2014. A Survey of Feature Selection and Feature Extraction Techniques in Machine Learning, *IEEE Science and Information Conference*, 372-378.

[2] Engel D., Hüttenberger L., and Hamann B. 2014. A Survey of Dimension Reduction Methods for High-dimensional Data Analysis and Visualization, *LNCS Springer*, 1-16.

[3] Pechenizkiy M., Puuronen S., and Tsymbal A. 2008. Feature Extraction for Classification in the Data Mining Process, *International Journal Information Theories & Applications*, 10 : 271-278.

[4] Delac, K., Grgic, M. and Grgic, S. 2005. Independent comparative study of PCA, ICA, and LDA on the FERET data set, *Int. J. Imaging Syst. Technol*, 15 : 252-260.

[5] Ramlee, R., Muda, A.K. and Syed Ahmad, S.S. 2013. PCA and LDA as dimension reduction for individuality of handwriting in writer verification, *International Conference on Intelligent Systems Design and Applications (ISDA)*, 104-108.

[6] Anaraki, J.R. and Eftekhari, M. 2013. Rough set based feature selection: A Review, *The 5th Conference on Information and Knowledge Technology*, Shiraz : 301-306.

[7] Galathiya, A.S., Ganatra, A.P. and Bhensdadia, C.K. 2012. Improved Decision Tree Induction Algorithm with Feature Selection, Cross Validation, Model Complexity and Reduced Error Pruning, *International Journal of Computer Science and Information Technologies*, 3, 2 : 3427-3431.

[8] Yager, R.R. 1989. On the Dempster-Shafer Framework and New Combination Rules, *Information Science*, 41, 2: 93-137.

[9] Mathon, B.R., Ozbek, M.M., and Pinder, G.F. 2010. Dempster–Shafer Theory Applied to Uncertainty Surrounding Permeability, *Mathematical Geosciences*, 42, 3 : 293-307.

[10] Guyon, I., Gunn, S., Nikravesh, M., and Zadeh, L.A. 2006. "Feature extraction: foundations and applications" (Studies in Fuzziness and Soft Computing), *Springer-Verlag*.

[11] Bencharef, O., Fakir, M., Minaoui, B., Hajraoui, A., and Oujaoura, M. 2012. Color objects recognition system based on artificial neural network with Zernike, Hu & Geodesic descriptors, *2012 6th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT)*, 338-343.

[12] Schubert, J. 2011. Conflict management in Dempster–Shafer theory using the degree of falsity, *International Journal of Approximate Reasoning*, 52, 3 : 449-460.

[13] Witten, I.H., Frank, E., Hall, M.A., and Pal, C.J. 2016. "Data Mining: Practical machine learning tools and techniques", *Morgan Kaufmann*.

[14] Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., and Alsaadi, F.E. 2017. A survey of deep neural network architectures and their applications, *Neurocomputing*, 234 : 11-26.

[15] Schmidhuber, J. 2015. Deep learning in neural networks: An overview, *Neural Networks*, 61 : 85-117.

[16] Krizhevsky, A., Sutskever, I., and Hinton, G.E. 2012. Imagenet classification with deep convolutional neural networks, *In Advances in neural information processing systems*, 1097-1105.

[17] Vedaldi, A. and Karel L. 2015. Matconvnet: Convolutional neural networks for matlab, Proceedings of the 23rd ACM international conference on Multimedia, *ACM*

[18] Chou T.N. 2017. Stock Trends Prediction Using the Feature Pattern Constructed with the Panel Data of Asian Stock Markets, *Proceedings of 2017 International Workshop on Computer Science and Engineering*, 1296-1301.