K-Nearest Neighbor Regression with Principal Component Analysis for Financial Time Series Prediction

Li Tang
School of Management and Economics
University of Electronic Science and Technology of China
Chengdu, China
86-18608046528
tangli@std.uestc.edu.cn

Heping Pan
Intelligent Finance Research Center
Chongqing Institute of Finance
Chongqing, China
86-13896169035
panhp@swingtum.com

Yiyong Yao
Tianfu College of Southwestern University of Finance and Economics
Chengdu, China
86-13980992337
yiyongyao@yahoo.com

ABSTRACT
This paper constructs an integrated model called PCA-KNN model for financial time series prediction. Based on a K-Nearest Neighbor (KNN) regression, a Principal Component Analysis (PCA) is applied to reduce redundancy information and data dimensionality. In a PCA-KNN model, the historical data set as input is generated by a sliding window, transformed by PCA to principal components with rich-information, and then input to KNN for prediction. In this paper, we integrate PCA with KNN that can not only reduce the data dimensionality to speed up the calculation of KNN, but also reduce redundancy information while remaining effective information improves the performance of KNN prediction. Two specific PCA-KNN models are tested on historical data sets of EUR/USD exchange rate and Chinese stock index during a 10-year period, achieving the best hit rate of 77.58%.

CCS Concepts
• **Computing methodologies → Feature selection**

Keywords
Principal component analysis; k-nearest neighbor; financial time series; prediction

1. INTRODUCTION
As the most essential task of financial market analysis, time series prediction has received serious research attention with comprehensive investigations. An effective analysis and prediction of financial market, especially of the financial time series has very practical significance, which can provide investors with essential information to support their decision-making and financial activities, A fair large of literatures have shown the probabilistic predictability of financial time series. The Autoregressive Integrated Moving Average (ARIMA) proposed by Box and Jenkins [1], [2], the Autoregressive conditional heteroskedasticity model (ARCH) proposed by Engle [3], [4], and the Generalized Autoregressive Conditional Heteroskedasticity (GARCH) proposed by Bollerslev [5], [7] assume the volatility of financial market prices may be predictable.

ICCAI 2018, March 12–14, 2018, Chengdu, China

ACM ISBN 978-1-4503-6419-5/18/03…$15.00
https://doi.org/10.1145/3194452.3194467

Moreover, from various perspectives, other cross-disciplinary researchers have pursued financial models, including typically the chaos-theoretical models of Farmer [8], [9], the models based on Support Vector Machine (SVM) of Vapnik [10], [11], the Neural Networks [12]-[15], and the K-Nearest Neighbor (KNN) prediction models [16]-[18].

Among the existing financial prediction models, Cover and Hart's KNN [19] is widely used as it can recall the similar instances intuitively from the feature space [17], [18]. However, the large amount of calculations involving ineffective information is a main disadvantage of KNN. A financial time series generated by a sliding window is usually a high-dimension series, thus it can amplify the disadvantage of KNN. In order to tackle this practical problem, we can apply the Principal Components Analysis (PCA) proposed by Pearson [20] in 1901 and developed by Hotlling [21] in 1933 which is an effective data dimension reduction method used commonly[22]-[24]. PCA is an orthogonal linear transformation that takes full account of the statistical characteristics of the variables, and effectively extracts the main information while eliminating the correlation among variables. Actually, the algorithms integrating PCA and KNN are used effectively in the field of signal processing [25], images analyzing [26], and recognition technology [27].

By the idea of extracting data features as input to KNN for recognition, this paper constructs a financial time series prediction called PCA-KNN model, integrating PCA for data dimension reduction and feature extraction, and KNN for generating prediction. We implement two specific PCA-KNN prediction models which are trained and tested on real historical data sets of EUR/USD exchange rate and Chinese stock index, achieving a best hit rate of 77.58%. A comparison between PCA-KNN models and KNN models shows the PCA-KNN models perform the best for the time being.

2. PCA-KNN PREDICTION MODEL
A PCA-KNN model consists of two parts: 1) PCA for reducing redundancy information and extracting essential features; and 2) KNN for financial prediction modeling.

2.1 A Frame Work of PCA-KNN Model
To start a PCA-KNN prediction model, we should first specify a time frame of the financial time series and assume a historical time series long enough to use. In this paper, we consider the

daily time frame and use a sliding window to cast a historical data set. For a financial price time series at period t, it comprises open price, high price, low price, close price and the volume. However, we only focus on the close price $C(t)$ in this paper. A relative return of the financial price is defined as

$$R(t)=\left(C(t)-C(t-1)\right)/C(t-1). \tag{1}$$

We assume a historical relative return time series exists, it can expressed as

$$\begin{aligned}&HisR(t,N)=\\&\left(R(t-(N-m)+1),R(t-(N-m)+2),\cdots,R(t)\right)\end{aligned}, \tag{2}$$

$HisR(t,N)$ is casted up to the time t using a sliding window with the width of m, so it also can be written as $HisR(t,m)\in HisR(t,N),\quad m\ll N$.

In a PCA-KNN model, $HisR(t,m)$ is first transformed through PCA to generate a set of principal components as the input to KNN for prediction. Thus a PCA-KNN prediction model can be defined as

$$R(t+1)=KNN\left\{PCA\left(HisR(t,m)\right),k\right\}, \tag{3}$$

where k and m are model parameters which means the number of k-nearest neighbors and the width of a sliding window, $m<<N$.

2.2 Principal Component Analysis (PCA) for Information Redundancy Reduction

According to equation (3), we can generate an input-output process $\mathbf{H}\rightarrow\mathbf{R}$, where

$$\mathbf{H}=\left(HisR(t-1,m)\quad\cdots\quad HisR\left(t-(N-m),m\right)\right)', \tag{4}$$

$$\mathbf{R}=\left(R(t)\quad R(t-1)\quad\cdots\quad R\left(t-(N-m)+1\right)\right)'. \tag{5}$$

Matrix $\mathbf{H}$ is the input matrix with high dimensionality and redundancy information. Generally, high dimensionality implies a large number of calculations and redundancy information affects the quality of prediction, in particular the stability and reliability.

PCA is a classical statistical analysis method which can transform data into a set of principal components with rich effective information while reducing the data dimensionality. So in order to improve the performance of prediction model, we can apply PCA to reduce the data dimensionality and redundancy information.

To apply PCA on the input matrix $\mathbf{H}$, the first step should be a normalization of $\mathbf{H}$

$$\mathbf{Z}^T=\left(normalization(\mathbf{H})\right)^T. \tag{6}$$

Then we can do a singular value decomposition of $\mathbf{Z}^T$

$$\mathbf{Z}^T=\mathbf{U\Sigma V}^T, \tag{7}$$

where matrix $\mathbf{U}$ is the eigenvector matrix of $\mathbf{Z}^T\mathbf{Z}$ and matrix $\mathbf{V}$ is an eigenvector of $\mathbf{ZZ}^T$ .Matrix $\mathbf{\Sigma}$ is a nonnegative rectangular diagonal matrix with a diagonal matrix $\mathbf{P}$ in its left part. And matrix $\mathbf{P}$ consists of eigenvalues $\sigma_i\,(i=1,2,\cdots,r)$ of $\mathbf{Z}^T\mathbf{Z}$.

A transformed data matrix $\mathbf{Y}$ can be obtained by

$$\mathbf{Y}=\mathbf{ZU}=\left(\mathbf{U\Sigma V}^T\right)^T\mathbf{U}=\mathbf{V\Sigma}^T. \tag{8}$$

The columns of matrix $\mathbf{Y}$ are composed of principal components sequentially. In general, the first p of the r principal components concentrate the most rich-information of the data matrix. Thus a new matrix $\mathbf{\Sigma}^*$ can be formed by the first p principle components

$$\mathbf{\Sigma^*}=\mathbf{I}_{p\times r}\mathbf{\Sigma}. \tag{9}$$

Correspondingly, a matrix $\mathbf{U}^*$ of lower dimensionality of matrix $\mathbf{U}$ can be calculated. And the matrix $\mathbf{Y}$ can be reduced as

$$\mathbf{Y}^*=\mathbf{ZU}^*=\mathbf{V}\left(\mathbf{\Sigma}^*\right)^T, \tag{10}$$

However, note that the new dimensionality p is a critical factor, and we can use an index called Cumulative Contribution Rate (CCR) which is usually required to be more than a threshold like 85% for help [5]

$$CCR_p=\left(\sum\nolimits_{i=1}^{p}\sigma_i\right)\Big/\left(\sum\nolimits_{i=1}^{r}\sigma_i\right), \tag{11}$$

Then a low-dimensional matrix $\mathbf{Y}^*$ can be input to KNN instead of matrix $\mathbf{H}$.

2.3 K-nearest Neighbor Regression for Financial Prediction

KNN is used for classification and regression widely as it is an effective nonparametric algorithm with intuitiveness and simplicity. In this paper, we adopt KNN regression for financial prediction. The low-dimensional matrix $\mathbf{Y}^*$ and data set $X(t)=HisR(t,m)$ are the input; the data set $X(t+1)=HisR(t+1,m)$ is the output for KNN. Thus the KNN prediction model can be expressed as

$$X(t+1)=KNN\left(X(t),\mathbf{Y}^*,k\right). \tag{12}$$

When we start a KNN algorithm, the similarity $S\left(X(t),X(i)\right)$ between data $X(t)$ and any data sample $X(i)(i=1,2,\cdots,p)$ in matrix $\mathbf{Y}^*$ should be calculated first

$$S\left(X(t),X(i)\right)=-\|X(t)-X(i)\|^2, \tag{13}$$

Sort S to find the first k max S and the k-nearest neighbors $X(j)(j=1,2,\cdots,k)$ and $k<p$. Then we can obtain the output $X(t+1)$

$$X(t+1)=\left(\sum\nolimits_{j=1}^{k}X(j)\right)\Big/k, \tag{14}$$

and also the $R(t+1)$ form $X(t+1) = HisR(t+1, m)$.

It should be noted that k is the key parameter in this model, so find the optimal k value is a practical problem. In general, different specific historical data set has different optimal k value which is usually found by experiments. Thus in this paper, we find each optimal k value for each specific model by experiments, setting the original k value as $k=1$, generating predictions based on k nearest neighbors and increasing k value with step length $\lambda = 1$. If k value continues to increase three times with no effect on improvement of prediction, we can stop and find the optimal k value makes the best performance of prediction.

2.4 Key Parameters and Performance Metrics of PCA-KNN Models

Equation (3) indicates that two structural parameters should be determined for a specific PCA-KNN model. One is m standing for the width of a sliding window for capturing the input data and another is k standing for the number of nearest neighbors to be used for prediction.

For evaluating the performance of the PCA-KNN prediction model, we use Hit Rate which can measure the accuracy of the prediction direction [28]. We express the real value of relative return as R^{real} and the predicted one as R^{pred}. The Hit Rate is defined as

$$HitRate = \left(\sum_{i=1}^{n} d(i)\right) \Big/ n \,, \qquad (15)$$

where $d(i) = 1$ when $R^{real} \times R^{pre.} > 0$ or $d(i) = 0$ when $R^{real} \times R^{pred} < 0$, and n is the number of samples.

3. TEST OF A SPECIFIC PCA-KNN MODEL

Floating exchange rate system has been used world-wide. It is very useful to perform trend prediction in order to avoid foreign exchange market risk. We a 10-year historical data set on foreign exchange market to train and test a PCA-KNN prediction model in this paper.

3.1 PK_EURUSDD1 for EUR/USD Exchange Rate Daily Returns Prediction

For predicting the $t+1$ daily return of the EUR/USD exchange rate, we propose a specific PK_EURUSDD1 model which can be expressed as

$$R(t+1) = KNN\left\{PCA\left(EURUSDD1\_HisR(t,m)\right), k\right\}. \qquad (16)$$

The specific historical data sets we used are the daily return of EUR/USD exchange rate spans over the time period of 2nd January 2007 to 24th November 2017 comprising 2830 trading days, in which the earlier 80% part is used for in-sample training and the later 20% part for out-of-sample testing.

We train the PK_EURUSDD1 model using the in-sample training data and test using the out-of-sample data. The test result in terms of hit rata are shown in Table 1. The best hit rate achieves 0.7556 (75.56%) with $m=15$ and $k=1$. For total 30 hit rates, 10 of them are higher than 0.70, 24 of them are higher than 0.65, and 100% of them are higher than 0.60. Thus it can be said that the PCA-KNN model is effectual for financial prediction on foreign exchange market.

Table 1. Performance of PK_EURUSDD1 predicting $R(t+1)$ with different window width m and k values

Window Width m	Hit Rate				
	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$
$m=10$	0.6242	0.6848	0.6343	0.6848	0.6444
$m=15$	**0.7556**	0.7152	0.7253	0.6545	0.6949
$m=20$	0.6545	0.6040	0.6646	0.7152	0.7051
$m=25$	0.6747	0.7051	0.7455	0.6747	0.6545
$m=30$	0.7455	0.6949	0.7152	0.6646	0.7253
$m=35$	0.6343	0.6646	0.6646	0.6646	0.6343

3.2 PK_HS300D1 for Chinese HS300 Index Daily Return Prediction

For predicting the $t+1$ daily return of Chinese HS300 index, we propose a specific PK_HS300D1 model which can be expressed as

$$R(t+1) = KNN\left\{PCA\left(HS300D1\_HisR(t,m)\right), k\right\}. \qquad (17)$$

The specific historical data sets we used is the daily return of HS300 index spans over the time period of 4th January 2007 to 28th July 2017 comprising 2571 trading days, in which the earlier 80% part is used for in-sample training and the later 20% part for out-of-sample testing.

We also train the PK_HS300D1 model using the in-sample training data and test using the out-of-sample data. The test result in terms of hit rate is shown in Table 2. The best hit rate achieves 0.7758 (77.58%) with $m=30$ and $k=2$. From the test results, it can be said that PK_HS3001 is an effective model for predicting the daily return of HS300 index.

Table 2. Performance of PK_HS300D1 predicting $R(t+1)$ with different window width m and k values

Window Width m	Hit Rate				
	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$
$m=10$	0.6444	0.6040	0.5838	0.6040	0.5939
$m=15$	0.7354	0.7354	0.7556	0.7253	0.7556
$m=20$	0.7457	0.7455	0.7152	0.7152	0.7657
$m=25$	0.6848	0.6646	0.7152	0.7455	0.7253
$m=30$	0.7253	**0.7758**	0.6646	0.7556	0.6949
$m=35$	0.7354	0.6747	0.7152	0.7051	0.7253

3.3 Comparison with KNN Model

For further testing, we construct two specific prediction models (without PCA) and compare their performances with the performance of PK_EURUSDD1 and PK_HS300D1 prediction models. The two KNN prediction models are KNN_EURUSDD1 and KNN_HS300D1

$$R(t+1) = KNN\{EURUSDD1\_HisR(t,m),k\}. \quad (18)$$

$$R(t+1) = KNN\{HS300D1\_HisR(t,m),k\}. \quad (19)$$

Correspondingly, these two models are trained and tested using the same in-sample and out-of-sample data as the PCA-KNN models. Table 3 shows the test result of KNN_EURUSDD1. The best hit rate is 0.6859 (68.59%) with $m = 20$ and $k = 1$. It is lower than the best hit rate of PK_EURUSDD1 model. Table 4 shows the test result of KNN_HS300D1. The best hit rate achieves 0.6963 (69.63%) with $m = 30$ and $k = 5$. This hit rate is also lower than the best hit rate of PK_HS300D1. Moreover, in both Table 3 and Table 4, only 50% of the hit rates are higher than 0.60, even none is higher than 0.70. In order to do a further comparison, we choose the best hit rate of each model to compare. Table 5 shows the comparison results. For the time being, the PCA-KNN models perform the best in terms of hit rate. In general, it can be said that PCA-KNN prediction model is a more effective one than KNN prediction model. Moreover, it implies PCA is useful in reducing redundancy information and improving the performance of KNN.

Table 3. Performance of KNN_EURUSDD1 predicting $R(t+1)$ with different window width m and k values

Window Width m	Hit Rate				
	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$
$m=10$	0.5848	0.6051	0.6051	0.6162	0.6061
$m=15$	0.5545	0.5747	0.5343	0.5747	0.6152
$m=20$	**0.6859**	0.6051	0.6051	0.5747	0.6051
$m=25$	0.6152	0.6253	0.5949	0.6051	0.6051
$m=30$	0.5848	0.6051	0.5646	0.5343	0.5040
$m=35$	0.5444	0.6859	0.5949	0.5242	0.5040

Table 4. Performance of KNN_HS300D1 predicting $R(t+1)$ with different window width m and k values

Window Width m	Hit Rate				
	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$
$m=10$	0.5939	0.5838	0.5737	0.6556	0.6657
$m=15$	0.5657	0.6263	0.6263	0.6263	0.6364
$m=20$	0.6253	0.5949	0.4737	0.4737	0.4737
$m=25$	0.5848	0.5848	0.6758	0.5848	0.5141
$m=30$	0.6152	0.5545	0.6051	0.6253	**0.6963**
$m=35$	0.6162	0.6263	0.6263	0.6263	0.6263

Table 5. Best performance comparison of PCA-KNN with KNN for prediction

Specific Model		Hit Rate
EUR/USD exchange rate	**PK_EURUSDD1**	**0.7556**
	KNN_EURUSDD1	0.6859
HS300 index	**PK_HS300D1**	**0.7758**
	KNN_HS300D1	0.6963

4. CONCLUSION

In this paper, we have proposed an integrated model PCA-KNN for financial prediction. A PCA-KNN model is composed of two technical components: PCA for reducing the data dimensionality and redundancy information while remaining principal components with rich-information, and KNN for financial prediction. Especially, PCA transforms the input data to a set of principal components as input for prediction that can reduce the calculation and improve the performance of KNN. Two specific KNN-PCA models are tested on EUR/USD exchange rate and Chinese HS300 index for predicting the daily returns, achieving a best hit rate of 0.7758 (77.58%). Also a comparison between KNN-PCA models and KNN models is implemented, resulting the PCA-KNN is a more effective model for financial prediction.

As a prediction model for financial time series, PCA-KNN comprises two key processes, feature extraction and modeling of prediction which are also the main factors for the performance of prediction. Thus in order to improve the prediction model, we can focus on these two procedures in our future work. For feature extraction, we can advance four aspects on: 1) taking more comprehensive information such as the open price, high price, low price and volumes as original input, 2) applying other effective nonlinear dimension reduction algorithm instead of the linear PCA, which should be more suitable for financial time series, 3) finding a similarity metric to take place of Euclidean measurement which is special for financial time series, 4) improving KNN as a weighted KNN with PCA loading as weights.

5. ACKNOWLEDGMENTS

We are grateful to the China National Science Foundation (Grant: 17BGL231) for financial research funding. This paper has benefited greatly from discussions with the participants of the IEEE/SICE International Symposium on System Integration 2017 Conference, Taipei, Taiwan.

6. REFERENCES

[1] Box, G. Jenkins, G. 1970. *Time Series Analysis: Forecasting and Control*. San Francisco: Holden-Day.

[2] Zhang, G. S., Zhang, X. D., and Feng, H. Y. 2016. Forecasting Financial Time Series Using A Methodology Based on Autoregressive Integrated Moving Average and Taylor Expansion. *Expert Systems*. 55 (Oct. 2016), 501-516. DOI= https://doi.org/10.1111/exsy.12164.

[3] Engle, R.F. 1982. Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation. *Econometrica*. 50 (Jul. 1982), 987-1007. DOI= https:// doi.org/10.2307/1912773.

[4] Davidson, J. and Li, X. Y. 2016. Strict Stationarity, Persistence and Volatility Forecasting in ARCH Process. *Journal of Empirical Finance*. 38 (Sep. 2016), 534-547. DOI= https://doi.org/10.1016/j.jempfin.2015.08.010.

[5] Bollerslev, T. 1986. Generalized Autoregressive Conditional Heteroskedasticity. *Journal of Econometrics*. 31 (Apr. 1986), 307-327. DOI= https:// doi.org/10.1016/0304-4076(86)90063-1.

[6] Beckers, B., Herwartz, H., and Seidel, M. 2017. Risk Forecasting in (T) GARCH Models with Uncorrelated Dependent Innovations. *Quantitative Finance*. 17 (Jan. 2017), 121-137. DOI= https://doi.org/10.1080/14697688.2016.1184303.

[7] Pedro, C. S. and Pedro, H. M. 2017. Volatility Forecasting via SVR-GARCH with Mixture of Gaussian Kernels. *Computational Management Science*. 14 (April 2017), 179-196. DOI= https://doi.org/10.1007/s10287-016-0267-0.

[8] Farmer, J.D. and Sidorowich, J.J. 1987. Predicting Chaotic Time Series. *Physical Review Letters*. 59 (Aug. 1987), 845-848. DOI= https://doi.org/10.1103/PhysRevLett.59.845.

[9] Ravi, V., Pradeepkumar, D., and Deb, K. 2017. Financial Time Series Prediction Using Hybrids of Chaos theory, Multi-layer Perceptron and Multi-objective Evolutionary Algorithms. *Swarm and Evolutionary Computation.* 36 (Oct. 2017), 136-149. DOI= https://doi.org/10.1016/j.swevo.2017.05.003.

[10] Vapnik,V. 1995. *The Nature of Statistical Learning Theory*. Hong Kong: Springer, New York.

[11] Sermpinis, G., Stasinakis, C., Theofilatos, K., and et al. 2015. Modeling, Forecasting, and Trading the EUR Exchange Rates with Hybrid Rolling Genetic Algorithms-Support Vector Regression Forecast Combinations. *Harvard Business Review*. 247 (Dec. 2015), 831-846. DOI= https://doi.org/10.1016/j.ejor.2015.06.052.

[12] Jena, P.R., Majhi, R., and Majhi, B. 2015. Development and Performance Evaluation of A Novel Knowledge Guided Artificial Neural Network (KGANN) Model for Exchange Rate Prediction. *Journal of King Saud University-Computer and Information Sciences*.27 (Oct. 2015), 450-457. DOI= https:// doi.org/10.1016/j.jksuci.2015.01.002.

[13] Shen, F., Chao, J., and Zhao, J. 2015. Forecasting Exchange Rate Using Deep Belief Networks and Conjugate Gradient Method. *Neurocomputing*.167 (Nov. 2015), 243-253. DOI= https:// doi.org/10.1016/j.neucom.2015.04.071.

[14] Hussain, A.J., Al-Jumeily, D., Al-Askar, H., and et al.2016. Regularized Dynamic Self-organized Neural Network Inspired by the Immune Algorithm for Financial Time Series Prediction. *Neurocomputing*.188 (May 2016), 23-30. DOI=https: //doi.org/10.1016/j.neucom.2015.01.109.

[15] Galeshchuk, S. 2016. Neural Networks Performance in Exchange Rate Prediction. *Neurocomputing*.172 (Jan 2016), 446-452. DOI=https://doi.org/10.1016/j.neucom.2015.03.100.

[16] Teixeira, L.A. and Oliveira, A.L.I. 2010. A Method for Automatic Stock Trading Combining Technical Analysis and Nearest Neighbor Classification. *Expert Systems with Applications* .37 (Oct. 2010), 6885-6890. DOI= https://doi.org/10.1016/j.eswa.2010.03.033.

[17] Zhang, N.N., Lin, A.J., and Shang, P.J. 2017. Multidimensional K-nearest Neighbor Model Based on EEMD for Financial Time Series Forecasting. *Physica A.* 477 (Jul. 2017), 161-173. DOI=https://doi.org/10.1016/j.physa.2017.02.072.

[18] Lin, A.J., Shang, P.J., Feng, G.C., and Zhong, B. 2012. Application of Empirical Mode Decomposition Combined with K-nearest Neighbors Approach in Financial Time Series Forecasting. *Fluctuation and Noise Letters*.11 (Nov. 2012), 1-14. DOI=https://doi.org/10.1142/S0219477512500186.

[19] Cover, T. and Hart, P. 1967. Nearest Neighbor Pattern Classification. *IEEE Transactions on Information Theory*. 13 (Jan. 1967), 21-27. DOI=https://doi.org/10.1109/TIT.1967.1053964.

[20] Karl Pearson, F.R.S. 1901. On Lines and Planes of Closest Fit to Systems of Points in Space. *Philosophical Magazine.* 2 (1901), 559-572. DOI=https://doi.org/10.1080/14786440109462720.

[21] Hotelling, H. 1933. Analysis of a Complex of Statistical Variables into Principal Components. *Journal of Educational Psychology.* 24 (1933), 417-441. DOI=https://doi.org/10.1037/h0071325.

[22] Li, Q. and Quan, H. 2014. The Dimension Reduction Method of Face Feature Parameters Based on Modular 2DPCA and PCA. *Applied Mechanics and Materials.687-691* (Nov. 2014), 4037-4041. DOI= https://doi.org/10.4028/www.scientific.net/AMM.687-691.4037.

[23] Haruo, H. and Aapo, H. 2016. Learning Visual Spatial Pooling by Strong PCA Dimension Reduction. *Neural Computation.* 28 (July 2016), 1249-1264. DOI= https://doi.org/10.1162/NECO_a_00843.

[24] Tu, Y., Hung, Y. S., Hu, L., and Zhang, Z. 2015. PCA-SIR: A New Nonlinear Supervised Dimension Reduction Method with Application to Pain Prediction from EEG. *2015 7th International IEEE/EMS Conference on Neural Engineering* (April 22 - 24, 2015). IEEE, Montpellier, France, 1004-1007. DOI= https://doi.org/10.1109/NER.2015.7146796.

[25] Gupta, V. and Mittal, M. 2018. KNN and PCA Classifier with Autoregressive Modeling during Different ECG Signal Interpretation. *Procedia Computer Science.* 125 (2018), 18-24. DOI= https://doi.org/10.1016/j.procs.2017.12.005.

[26] Justiawan, Riyanto, S., and Zainal, A. 2017. Tooth Color Detection Using PCA and KNN Classifier Algorithm Based on Color Moment. *Emitter: International Journal of Engineering Technology.* 5 (July 2017), 139- 153. DOI= https://doi.org/10.24003/emitter.v5i1.171.

[27] Wang, Q., Jia, K., and Liu, P. 2015. Design and Implementation of Remote Facial Expression Recognition Surveillance System Based on PCA and KNN Algorithms. *2015 International Conference on Intelligent Information Hiding and Multimedia Signal Processing* (September 23 - 25, 2015). IEEE, Adelaide, SA, Australia, 314-317. DOI= https://doi.org/10.1109/IIH-MSP.2015.54.

[28] Pan, H.P., Haidar, I., and Kulkarni ,S. 2009. Daily Prediction of Short-Term Trends of Crude Oil Prices Using Neural Networks Exploiting Multimarket Dynamics. *Frontiers of Computer Science in China.* 3 (June 2009), 177-191. DOI=https://doi.org/10.1007/s11704-009-0025-3.