Machine Learning Classification Techniques to Predict Directional Change of Energy Prices Using High Dimensionality Reduction

Vidya Moni
Dept of Electrical Engineering
National Institute of Technology Warangal
Warangal, India
vm831871@student.nitw.ac.in

Maheshwari Mattipalli
Dept of Electrical Engineering
National Institute of Technology Warangal
Warangal, India
mm841815@student.nitw.ac.in

Altaf Q. H. Badar
Dept of Electrical Engineering
National Institute of Technology Warangal
Warangal, India
altafbadar@nitw.ac.in

2022 International Conference on Computer Science and Software Engineering (CSASE) | 978-1-6654-2632-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/CSASE51777.2022.9759655

***Abstract*—In today's economy, energy is an essential commodity. Every individual and business use energy for their needs. Electricity is the most common form in which energy is consumed. Thus, accurately predicting electricity prices could aid businesses in planning their finances and logistics and have a better long-term vision of their company. In this paper, the next-day directional change of the electricity prices of the German and Austrian areas of the European Energy Exchange (EEX) wholesale market is predicted based on several parameters, including the daily Phelix index, the volume of trade, coal prices, Title Transfer Facility (TTF), wind power production, and many others. High Dimensionality Reduction techniques (Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA)) are used in conjunction with Machine Learning (ML) classification algorithms; Support Vector Machines (SVM), and Artificial Neural Networks (ANN), in particular. The software employed for this research was Python 3, used on Google Collaboratory. The maximum forecast accuracy achieved by our model was 75.00%.**

Keywords—Machine Learning, Support Vector Machines, Artificial Neural Networks, Principal Component Analysis, Linear Discriminant Analysis, Energy Price

I. Introduction

In the 1990s, electricity started being considered a commodity [1]. This caused the transition from regulated regional electricity monopolies (typically controlled by governments) to unregulated private electric energy markets. The European Energy Exchange (EEX) was established in 2002. It offers a platform for trading energy, natural gas, freight, and metals, to stabilize and regularize energy trading in free markets [2].

The Phelix day base and Phelix day peak power indices represent the power derivatives market in the German/Austrian region in this paper. The day peak refers to the peak hours of the power grid, which are anticipated to be between 8:00 AM and 8:00 PM. The baseload relates to the overall demand on the power grid daily. Power market behavior is influenced by the unique characteristics of electricity usage, such as multiple seasonality, the calendar impact (consumption behavior on weekends and holidays), high volatility, and non-stationarity. These qualities combine to create a complicated environment in which electricity price prediction is a challenging undertaking [3].

This paper uses SVM and ANN in conjunction with High Dimensionality Techniques (PCA and LDA) to anticipate the directional change in electricity prices in the EEX wholesale market for the German and Austrian regions. Daily-average spot price delays (the dependent variable) are utilized as input variables, along with other critical explanatory factors such as liquid and solid fuel costs and the total volume-of-trade of electric energy. After a thorough search for the optimal input variables and parameter values, the best classification model was chosen.

The paper is organized in the following manner: In Section II, there is a brief discussion about ML, followed by a description of the ML algorithms used in the research. Section III explains High Dimensionality Reduction and elaborates upon PCA and LDA. Section IV provides details related to the dataset. Section V describes the methodology, while Sections VI and VII provide the Results and Discussion, the conclusion, and future work.

II. Machine Learning and Deep Learning

ML is a cutting-edge approach to problem-solving in prediction, classification, clustering, and modeling. It is a subset of Artificial Intelligence (AI) that has exponentially increased in prominence during the last few decades. The basic premise of ML is that a computer, or machine, "learns" from data provided to it, i.e., finds patterns and correlations in a dataset to accurately forecast or model the output. As more data is fed into the computer, it discovers more insights, improving the accuracy of the output. Unlike prior optimization techniques, most ML models do not contain coded rules. ML often employs implicit learning and develops its own modeling rules by minimizing the difference between the expected and true outputs (ground truth) until saturation. For decades, ML has been used to analyze massive datasets and uncover and extract patterns from within the data.

Deep learning is a subset of a large family of ML techniques based on representation learning and ANN [4]. There are three types of learning: supervised, semi-supervised, and unsupervised. Deep-learning architectures such as deep neural networks, deep belief networks, deep reinforcement learning,

recurrent neural networks, and convolutional neural networks have been used in fields such as computer vision, speech recognition, natural language processing, machine translation, bioinformatics, drug design, medical image analysis, material inspection, and board game programs, with results comparable to, and in some cases exceeding, human performance [5].

The current problem statement is classification. Several state-of-the-art ML algorithms are extensively used for classification, including Naive-Bayes, Decision Trees, Random Forests, SVM, ANN, Logistic Regression, and many others [6]. There have been many applications of ML techniques to energy forecasting in recent years. Some of these will be explained in the next section.

A. Machine Learning Applied to Energy Forecasting

There have been many approaches to energy price forecasting using statistical models and ML. Cuaresma *et al.* looked at the effectiveness of univariate models in projecting energy prices in the German market [7]. They employed two sets of autoregressive (AR) and autoregressive moving average (ARMA) models with various settings: single time series models and separate hour-by-hour time series models, to evaluate the forecasting abilities of univariate models. They concluded that the distinct time series ARMA models, which account for price spikes, outperform the single time series models and AR models using out-of-sample comparisons. In [8], Erlwein *et al.* suggested a hidden Markov model estimate pricing for the next day, considering buyer and seller bidding strategies, unforeseen weather fluctuations, and concerns with the manufacturing process. Pao presented an ANN for price forecasting in the EEX market over the long term. For three distinct periods, the authors compared the out-of-sample forecasting performance of an ANN with an autoregressive model. They found that the ANN model outperformed the autoregressive model [9].

B. Support Vector Machines

A common ML-based categorization technology, SVM, is widely utilized for a range of predictive analytics. An SVM is a supervised ML model for two-group classification problems based on the usage of classification techniques [10]. They would be able to categorize fresh data after training an SVM model with labeled data. Compared to newer algorithms such as neural networks, SVM is preferable because of its faster processing speed and performance with fewer samples [11]. SVM is most commonly employed as a binary classifier, where data is sorted into one of two classes: class 0 or class 1. A hyperplane is chosen in the vector machine to do this. A hyperplane is defined as a function that may participate in variable space [12]. The vector machine learning computation uses a hyperplane to identify the coefficients that result in the best separation of the classes. The procedure is known as SVM when the hyperplane takes on a linear shape. On the other hand, the hyperplane can assume any shape based on the best fit for the separation margin between two classes: polynomial, Gaussian, Radial Basis Function (RBF), Laplace RBF, or sigmoid function. Whenever the hyperplane takes on a nonlinear function, we call it Kernel SVM.

The linear SVM formulation is the simplest, with the hyperplane on the space of the input data x. The hypothesis space in this example is a subset of all hyperplanes of the form:

$$f(x) = w.x + b \qquad (1)$$

SVMs, in their most basic form, look for a hyperplane in a space other than the input data x. They look for a hyperplane produced by a kernel K in the feature space (the kernel defines a dot product in that space). The hypothesis space is defined as a set of "hyperplanes" in the feature space created by K. This space can formally be referred to as:

$$\{f : ||f||^2_K < \infty \}$$

Where K is the kernel that defines the RKHS, and ||f||2K is the function's Reproducing Kernel Hilbert Space (RKHS) norm. SVM consider subsets of this space, namely sets of the form:

$$\{f : ||f||^2_K \leq A^2\}$$

For some constant A, the constant A defines a hypothesis space structure (the larger A is, the more complex the hypothesis space is). SVM aims to discover the solution that has the "best" RKHS norm, i.e. the best A.

The next choice we must make is that of the loss function. The real loss function used for SVM classification is

$$|y - f(x)|_+$$

(that is, 0 if y - f(x) is 0, and y - f(x) otherwise). Here, y is the ground truth value of the output, and f(x) is the value predicted by the SVM. This is done to deal with scaling and computational concerns [13].

C. Artificial Neural Networks

ANN is a commonly used Deep Learning algorithm. ANNs may be used to solve problem statements centred on prediction and categorization. The neuron in the human brain is imitated in this Deep Learning model, and an elaborate lattice of such neurons is formed, capable of extraordinarily high performance. A backpropagation network, which operates on a multi-layer, feed-forward neural network, is the most often used ANN algorithm [14]. The input, hidden, and output layers are the three layers that neurons can have. The feature-designed inputs from the dataset being worked on make up the input layer of an ANN. The output layer contains the outputs needed to solve the research challenge. The hidden layers are layers of neurons that help the model's performance but are the neurons' weights, as determined by the backpropagation process. Recurrent Neural Network (RNN), Convolutional Neural Network (CNN), Long Short Term Memory Network (LSTM), and the Multilayer Perceptron (MLP) are some more neural network architectures.

Let a single input be defined as xj. An artificial neuron (represented by ai) passes its inputs through a set of synaptic weights (represented by wij), which get adjusted as the algorithm iterations go on. The synaptic weights adjust themselves based on the loss function to minimize it. The artificial neuron computes the weighted sum of its inputs, to which it adds a bias (represented by bi). This value is then passed through an activation function, defined as *f*. The final output of each neuron is shown in Equation 2:

$$a_i = f\left(\sum_{j=1}^{n} w_{ij} x_j + b_i\right) \qquad (2)$$

The structure of an artificial neuron is given in Fig. 1:

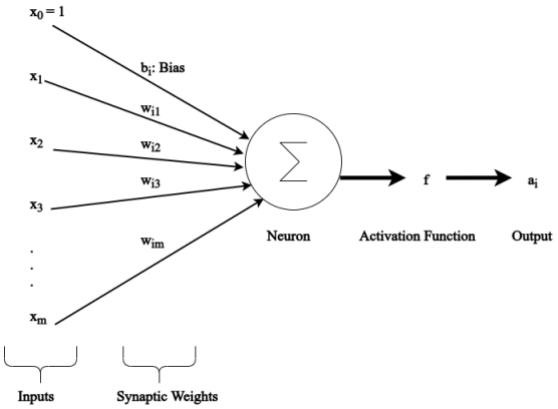

Fig 1. Structure of a Standard Artificial Neuron

A simple, feed-forward multi-layer neural network is depicted in Figure 2:

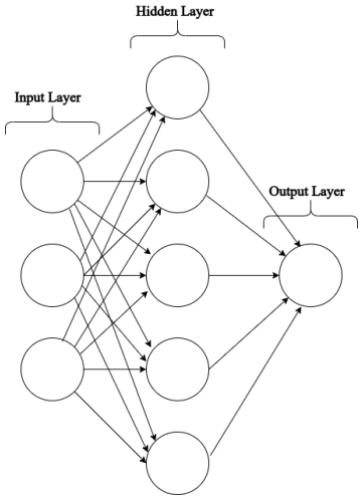

Fig 2. Simple Feed-Forward Neural Network

Between Figs. 1 and 2, the whole structure of an ANN can be visualized. Three sorts of parameters are commonly used to define an ANN [15]:

1. The pattern of connectivity between the several layers of neurons;
2. The learning procedure for changing the interconnectivity weights;
3. The activation function is used to transform the weighted input of a neuron into its output.

III. High Dimensionality Reduction

The most fundamental and widely used approach for removing unnecessary characteristics from datasets is dimensionality reduction. It is primarily divided into two sub-categories: feature extraction and feature selection. Numerous features are combined to create a new feature with a smaller feature area using the feature extraction approach. PCA, Canonical Correlation Analysis (CCA), and LDA are examples of feature extraction techniques [16]. On the other hand, the feature selection strategy selects a subset of features from the dataset to feature redundancy and increases feature relevance to the target class label. Chi-square [17], Fisher score [18], Gain [19], Relief [20], and minimal redundancy and maximum relevance [21] are some examples of feature selection algorithms. Feature extraction and feature selection strategies can increase learning performance in accuracy, model interpretability, computational complexity, and storage needs. In terms of interpretability and readability, feature selection is preferred over feature extraction. Maintaining the original characteristics in the subset resulting from feature selection is vital in various study domains, such as finding relevant genes to target diseases in the medical sector [22].

A. Principal Component Analysis

In several fields, large datasets are becoming more common. To analyze large datasets, approaches must dramatically decrease their dimensionality in an interpretable manner. This preserves the majority of the data's content. Various techniques have been developed for this goal, but PCA is one of the oldest and most commonly utilized. Its goal is to minimize a dataset's dimensionality while keeping as much 'variability' (i.e. statistical information) feasible [23].

Simply speaking, there are six steps to conduct PCA on a dataset. These are as follows [24]:

1. Take the entire dataset with *d+1* dimensions and remove the labels, resulting in a d dimensional dataset.
2. Calculate the mean for each of the dataset's dimensions.
3. Compute the entire dataset's covariance matrix.

$$cov(X,Y) = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{x})(Y_i - \bar{y}) \qquad (3)$$

4. Calculate the eigenvalues for each eigenvector. The eigenvalues of A are the roots of the characteristic equation given below:

$$\det(\boldsymbol{A}\text{-}\lambda\mathrm{I}) = 0$$

5. To build a $d \times k$ dimensional matrix W, we need to sort the eigenvectors by decreasing eigenvalues and pick k eigenvectors with the most significant eigenvalues.
6. Transform the data onto the new subspace using this $d \times k$ eigenvector matrix.

B. Linear Discriminant Analysis

LDA is a robust classification approach as well as a dimension reduction tool. We can get the same LDA features with or without the data normality assumption, which explains its resilience [25]. LDA is a technique for classifying, reducing dimensions, and visualizing data. It has been in existence for quite some time. LDA frequently yields reliable, reasonable, and understandable classification results [26]. When it comes to real-world classification challenges, LDA is commonly used as a first and benchmarking approach before moving on to more intricate and flexible methods.

Simply speaking, there are five steps to conduct PCA on a dataset. These are as follows:

1. Calculate the d-dimensional mean vectors for each of the dataset's classes.
2. The scatter matrices must be computed (in-between-class and within-class scatter matrices).
3. Calculate the scatter matrices' eigenvectors (e1, e2, ..., ed) and corresponding eigenvalues (λ1, λ2, ..., λd).
4. To build a d × k dimensional matrix W, sort the eigenvectors by decreasing eigenvalues and choose k eigenvectors with the largest eigenvalues (where every column represents an eigenvector).
5. This d × k eigenvector matrix can now be used to transform the samples onto the new subspace. Let X be an n × d dimensional matrix representing the n samples and let Y be the transformed matrix in the new space of n × k dimensions. This step can be summarized as the matrix multiplication, Y = X × W.

IV. Dataset Collection

The dataset can be found in [3]. There were a total of sixty input variables, including the Phelix spot lag, Phelix volume, various power indices (Amsterdam Power Exchange (APX), Zeebrugge Gas Index, Net Connect Germany (NCG), Brent Index), coal prices in Rotterdam and Richards Bay, and wind power consumed by Germany, to name a few. The output was the directional change of the electricity price (1 for an increase in price compared to the previous day and 0 for a decrease). The data were considered from 1 October 2008 to 22 Feb 2013.

An exploratory data analysis found that increased power indices directly correlated with increased electricity prices. Similarly, an increase in coal prices also showed a corresponding rise in electricity prices. We also found that an increase in the use of wind power caused a decrease in electricity prices. However, the correlation between the Phelix index and the directional change in electricity prices was ambiguous. Similarly, every input feature was analyzed with respect to the output, and the features which showed the highest correlation were considered in the final modified dataset.

V. Methodology

This section presents our approach for using ML classification techniques to predict the directional change of the electricity price for German and Austrian areas in the EEX market using High Dimensionality Reduction, followed by SVM and ANN.

The data was cleaned and pre-processed. An exploratory data analysis was conducted, where we analyzed which of the input features had the most significant correlation with the output. The most relevant components were selected, thereby reducing the size of the dataset to reduce the computational power required. We selected fifty out of the sixty input features present in the original dataset based on the correlation between the inputs and the output class. First, SVM and ANN techniques were directly used to predict the directional change in the electricity price. As the number of parameters was quite significant, High Dimensionality Reduction techniques were used to reduce the dataset's size further. First, PCA was performed, followed by SVM and ANN. We then performed LDA on the dataset, followed by SVM and ANN.

The dataset was split into a training set and a testing set to perform SVM and ANN. The data was divided using Python 3's randomization functionality, with 80% of the data going to the training set and 20% to the test set. The algorithms were run with the help of a probabilistic binary classifier. These algorithms classify any hypothesis with a probability of 0.5 or higher as positive and any hypothesis with a probability of less than 0.5 as negative.

In the case of ANN, there were more than a hundred structures of neural networks considered, with up to ten hidden layers, with varying numbers of neurons in each layer. The final structure, which gave us the best result, contained four hidden layers. The first layer contained eight neurons with a Rectified Linear Unit (ReLU) activation function. The second layer contained ten neurons with a Sigmoid activation function. The third layer contained six neurons with a Sigmoid function. The final hidden layer had the same structure as the second hidden layer. The Adam optimizer was used, along with the binary cross-entropy loss function. The accuracy provided by this neural network was 66.84%.

Similarly, with PCA, experimentation was done regarding the number of principal components. The results were similar, and it was shown that PCA was not a good high dimensionality reduction approach for this dataset. After trying various combinations, the best hyperplane shape for SVM was linear. We obtained the confusion matrix for each result. Based on these matrices, the accuracies of the various combinations were calculated.

VI. Results and Discussion

To calculate the goodness of prediction, the following exactness measures were used:

1. Accuracy: It is defined as the sum of the True Positives (TP) and True Negatives (TN) divided by the sum of TP, TN, False Positives (FP), and False Negatives (FN).

2. Precision is defined as the fraction of TP to the sum of TP and FP, i.e., the total number of positive values predicted.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

3. The recall is defined as the fraction of TP to the sum of the TP and FN.

$$Recall = \frac{TP}{TP + FP}$$

The results of our research are summarized in the Table below:

TABLE I. ACCURACY, PRECISION, AND RECALL OF COMBINATIONS CONSIDERED

Combination Of High Dimensionality Reduction Technique And ML Classifier	Accuracy	Precision	Recall
SVM without High Dimensionality Reduction	71.70%	76.67%	87.12%
SVM with PCA	71.70%	71.74%	100%
SVM with LDA	75.00%	83.59%	81.06%
ANN without High Dimensionality Reduction	66.84%	73.84%	78.05%
ANN with PCA	66.84%	68.02%	95.12%
ANN with LDA	66.30%	75.89%	69.11%

The most common metric to evaluate the goodness of an ML classifier is accuracy. Thus, in the following figures, we graphically represented the accuracy comparison of each combination considered.

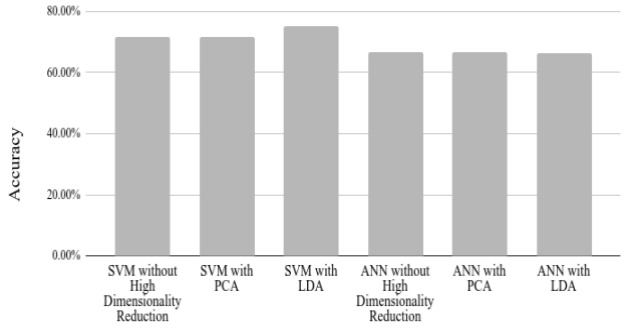

Fig 3. Accuracy of the Various Combinations Considered

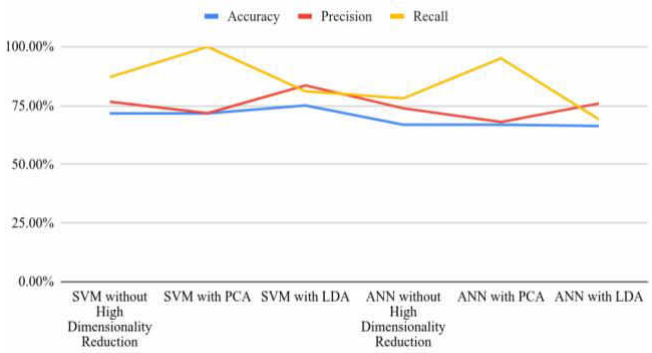

Fig 4. Comparison of Accuracy, Precision, and Recall

Table I can conclude that SVM was a better classification technique for this dataset than ANN. PCA did not improve the accuracy of the SVM and ANN classifiers. It reduced the precision of the two classifiers but improved the recall significantly. As for LDA, it performed excellently with SVM, increasing the accuracy significantly, from 71.70% to 75.00%, as shown in Fig. 3. It also improved the precision of the SVM classifier and reduced the recall by a small amount. With ANN, LDA reduced the accuracy by a minuscule amount, improved the precision slightly, and reduced the recall significantly. From Fig. 4, we can say that SVM, in conjunction with LDA, performed the best overall combinations.

PCA did not affect the accuracy of the classifiers. Upon further investigation, we found that a possible reason for this could be that PCA is strongly aligned to the input parameters with the highest variance, not necessarily those with the highest correlation to the output. It is an unsupervised algorithm and does not consider the labels of the training data during the transformation. LDA performed well in conjunction with SVM. Being a supervised algorithm, it considers the labels of the training data during the transformation. When there is a relatively small amount of data, PCA outperforms LDA. However, our dataset was relatively large; LDA outperformed PCA.

As one of the most widely used ML-based classifications approaches, SVM has many applications. Due to their more extraordinary performance versus limited data, SVMs outperform newer ML algorithms. Because anomalous data is given a lesser "say" in prediction based on how big of an outlier it is, the usage of a hyperplane provides great prediction accuracy. The total amount of the subtleties achieved by Deep Learning has not been realized due to the essentially linear form of the data since most synaptic connections in the network were straightforward.

VII. CONCLUSION AND FUTURE WORK

Various neural network structures, principal components, and hyperplane functions were explored. The best results were obtained from an ANN with four hidden layers, with the first layer containing eight neurons with a ReLU activation function, the second layer containing ten neurons with a Sigmoid activation function, the third layer containing six neurons with a Sigmoid function, and the final hidden layer having the same structure as the second hidden layer. The accuracy provided by this neural network was 66.84%. SVM outperformed ANN, giving an accuracy of 71.70%. The best function to use for the SVM hyperplane was linear. PCA did not affect the accuracies of the ML classifiers, although it did increase the recall in both cases. LDA performed best with SVM and slightly reduced the ANN classifier's accuracy. From Table I and Fig. 4, the best combination of ML classifier and High Dimensionality Reduction algorithm was SVM and LDA, giving the highest accuracy of 75.00%, with a precision of 83.59% and a recall of 81.06%. SVM significantly outperformed ANN in all three combinations, so we conclude that SVM is a better classification algorithm than ANN for this volume of data.

Future work will include exploring more High Dimensionality Reduction techniques, ML algorithms, and genetic algorithms to classify the direction of change in electricity prices in the German and Austrian regions of the EEX market.

REFERENCES

[1] P. Joskow, "Lessons Learned from Electricity Market Liberalization", *The Energy Journal*, vol. 29, no. 01, 2008. Available: 10.5547/issn0195-6574-ej-vol29-nosi2-3.

[2] T. Nedev, "European Energy Exchange EEX - Alternative for the Development of the Bulgarian Electricity Market", *Academic Journal of Interdisciplinary Studies*, 2015. Available: 10.5901/ajis.2015.v4n1s2p259.

[3] T. Papadimitriou, P. Gogas and E. Stathakis, "Forecasting energy markets using support vector machines", *Energy Economics*, vol. 44, pp. 135-142, 2014. Available: 10.1016/j.eneco.2014.03.017.

[4] J. Fan, C. Ma and Y. Zhong, "A Selective Overview of Deep Learning", *Statistical Science*, vol. 36, no. 2, 2021. Available: 10.1214/20-sts783.

[5] K. Lee, "Study on Deep Learning: Applications and Research Trends", Journal of Advanced Research in Dynamical and Control Systems, vol. 12, no. 7, pp. 1603-1611, 2020. Available: 10.5373/jardcs/v12sp7/20202264.

[6] V. Moni and S. Cumarasurier, "Human Papillomavirus Targeted Immunotherapy Outcome Prediction Using Machine Learning", *International Journal for Research in Applied Science and Engineering Technology*, vol. 9, no., pp. 3598-3611, 2021. Available: 10.22214/ijraset.2021.37197.

[7] J. Crespo Cuaresma, J. Hlouskova, S. Kossmeier and M. Obersteiner, "Forecasting electricity spot-prices using linear univariate time-series models", *Applied Energy*, vol. 77, no. 1, pp. 87-106, 2004. Available: 10.1016/s0306-2619(03)00096-5

[8] C. Erlwein, F. Benth and R. Mamon, "HMM filtering and parameter estimation of an electricity spot price model", *Energy Economics*, vol. 32, no. 5, pp. 1034-1043, 2010. Available: 10.1016/j.eneco.2010.01.005

[9] H. Pao, "A Neural Network Approach to m-Daily-Ahead Electricity Price Prediction", *Advances in Neural Networks - ISNN 2006*, pp. 1284-1289, 2006. Available: 10.1007/11760023_186

[10] Z. Kun, T. Ying-jie and D. Nai-yang, "Unsupervised and Semi-Supervised Two-class Support Vector Machines," *Sixth IEEE International Conference on Data Mining - Workshops (ICDMW'06)*, 2006, pp. 813-817, doi: 10.1109/ICDMW.2006.164.

[11] S. REN, D. YANG, X. LI and Z. ZHUANG, "Piecewise Support Vector Machines", *Chinese Journal of Computers*, vol. 32, no. 1, pp. 77-85, 2009. Available: 10.3724/sp.j.1016.2009.00077.

[12] A. C. Braun, U. Weidner and S. Hinz, "Support vector machines, import vector machines and relevance vector machines for hyperspectral classification — A comparison," 2011 3rd Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), 2011, pp. 1-4, doi: 10.1109/WHISPERS.2011.6080861.

[13] Vapnik V.," Statistical Learning Theory", Wiley, New York, 1998.

[14] L. Ozyilmaz and T. Yildirim, "Artificial neural networks for diagnosis of hepatitis disease," Proceedings of the International Joint Conference on Neural Networks, 2003., 2003, pp. 586-589 vol.1, doi: 10.1109/IJCNN.2003.1223422.

[15] M. Mishra and M. Srivastava, "A view of Artificial Neural Network," 2014 International Conference on Advances in Engineering & Technology Research (ICAETR - 2014), 2014, pp. 1-3, doi: 10.1109/ICAETR.2014.7012785.

[16] P. Korhonen, "Subjective principal component analysis", *Computational Statistics & Data Analysis*, vol. 2, no. 3, pp. 243-255, 1984. Available: 10.1016/0167-9473(84)90016-1.

[17] H. Liu, R. Setiono, "Chi2: feature selection and discretization of numeric attributes", Tools with Artificial Intelligence, 1995. Proceedings., Seventh International Conference on, IEEE, 1995, pp. 388–391.

[18] R.O. Duda, P.E. Hart, D.G. Stork, "Pattern Classification", John Wiley & Sons, 1999.

[19] Q. -w. Meng, Q. He, N. Li, X. -r. Du and L. -n. Su, "Crisp Decision Tree Induction Based on Fuzzy Decision Tree Algorithm," *2009 First International Conference on Information Science and Engineering*, 2009, pp. 4811-4814, doi: 10.1109/ICISE.2009.440.

[20] I. Kononenko, "Estimating attributes: analysis and extensions of relief, in: Machine Learning", ECML-94, Springer, 1994, pp. 171–182.

[21] H. Peng, F. Long, C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy", IEEE Trans. Pattern Anal. Mach. Intell. 27 (8) (2005), pp. 1226–1238.

[22] V. Bolon-Canedo, N. Sanchez-Marono, A. Alonso-Betanzos, J.M. Benitez, F. Herrera, "A review of microarray datasets and applied feature selection Methods", Information Sciences 282 (2014) 111–135

[23] T. Chin and D. Suter, "Incremental Kernel Principal Component Analysis," in IEEE Transactions on Image Processing, vol. 16, no. 6, pp. 1662-1674, June 2007, doi: 10.1109/TIP.2007.896668.

[24] S. Sehgal, H. Singh, M. Agarwal, V. Bhasker and Shantanu, "Data analysis using principal component analysis," 2014 International Conference on Medical Imaging, m-Health and Emerging Communication Systems (MedCom), 2014, pp. 45-48, doi: 10.1109/MedCom.2014.7005973.

[25] S. Ji and J. Ye, "Generalized Linear Discriminant Analysis: A Unified Framework and Efficient Model Selection," in IEEE Transactions on Neural Networks, vol. 19, no. 10, pp. 1768-1782, Oct. 2008, doi: 10.1109/TNN.2008.2002078.

[26] Tianwei Xu, Chong Lu and Wanquan Liu, "The matrix form for weighted linear discriminant analysis and fractional linear discriminant analysis," 2009 International Conference on Machine Learning and Cybernetics, 2009, pp. 1621-1627, doi: 10.1109/ICMLC.2009.5212309.