International Conference on Intelligent Sustainable Systems (ICISS 2019)
IEEE Xplore Part Number: CFP19M19-ART; ISBN: 978-1-5386-7799-5

A Study on Support Vector Machine based Linear and Non-Linear Pattern Classification

Sourish Ghosh
School Of Computer Engineering
KIIT University, Bhubaneswar, India
1605407@kiit.ac.in

Anasuya Dasgupta
School Of Computer Engineering
KIIT University, Bhubaneswar, India
1605339@kiit.ac.in

Aleena Swetapadma
School Of Computer Engineering
KIIT University, Bhubaneswar, India
aleena.swetapadma@gmail.com

***Abstract*—The best way to acquire knowledge about an algorithm is feeding it data and checking the result. In a layman's language machine learning can be called as an ideological child or evolution of the idea of understanding algorithm through data. Machine learning can be subdivided into two paradigms, supervised learning and unsupervised learning. Supervised learning is implemented to classify data using algorithms like support vector machines (SVM), linear regression, logistic regression, neural networks, nearest neighbor etc. Supervised learning algorithm uses the concepts of classification and regression. Linear classification was earlier used to form the decision plane but was bidimensional. But a particular dataset might have required a non linear decision plane. This gave the idea of the support vector machine algorithm which can be used to generate a non linear decision boundary using the kernel function. SVM is a vast concept and can be implemented on various real world problems like face detection, handwriting detection and many more. This paper surveys the various concepts of support vector machines, some of its real life applications and future aspects of SVM.**

Keywords—Pattern analysis, SVM, Classification, Machine Learning.

I. INTRODUCTION

Support vector machine being an implementation of the supervised learning paradigm, mainly deals with the ideas of classification and regression. Vapnik and Chervonenkis invented the support vector machine [1, 2]. In early ages of SVM, it could only classify data linearly by drawing hyper-planes. Later in 1992, Vapnik, Boser and Guyon introduced a way of building a non linear classifier by using the Kernel function (kernel was introduced in a research paper by Vapnik and Cortes which was published in 1995). Since then SVM has been one of the most dominant classification algorithms implementing supervised learning i.e. datasets defined by features and class labels. Later, Vapnik and Siegelmann introduced SVM clustering to implement unsupervised learning i.e. datasets without class labels and output feature.

SVM mainly belongs to the supervised learning technique which is formulated as given in [19]. There are training data, $\{(x_1, y_1) \ldots (x_n, y_n)\}$ in R^n x R which is sampled according to a unknown probability distribution P(x, y) and a generic loss function V(y, f(x)) that calculates the error, for a given x. f(x) is predicted in place of the actual value of y [3]. A function has to be found that minimizes the expectation of error on the new data i.e. finding the function f that minimizes the error as given in (1),

$$\int V(y, f(x))P(x, y)\, dx\, dy \qquad -(1)$$

A supervised learning algorithm thus, analyzes the training data set and maps it onto a function which is further used for mapping new examples. Classification is the process by which an algorithm analyzes the decision boundary taking help from the training data containing known observations. The most common example of classification would be detection of spam messages. The most general type of classification which is used is known as binary classifications where there is a need for discretise the outputs between 1 or 0. Thus a general decision boundary is required as given below.

$h_\Theta(x) \geq 0.5 \rightarrow y=1$

$h_\Theta(x) < 0.5 \rightarrow y=0$

Here 0.5 is the decision boundary. Also used is regression analysis, which is a statistical process for estimating the relationships among variables. It has many techniques for analyzing several variables, and to build relationship between a dependent variable and one or more independent variables [3, 4]. Here the regression function is estimated which is the most important aspect is the curve fitting concept. This uses the method of estimation of cost function from the training data as given in (2),

$$J(\Theta_0, \Theta_1) = \frac{1}{2m}\sum_{i=1}^{m}(h_\Theta(x_i) - Y_i)^2 \qquad -(2)$$

Where J= squared error cost function on Θ_0 and Θ_1, m= number of training examples, i= , Θ_0 = parameters for the hypothesis h_Θ , Θ_1 = parameters for the hypothesis h_Θ , h_Θ = predicted value of the hypothesis, x_i = independent feature, Y_i = actual value to find the minimized error function.

Earlier with use of linear classification and regression techniques the algorithms could be trained. But later it has been seen that a particular data set may require a non linear decision boundary in cases as shown in Fig. 1. This led to the rise of support vector machines. In the above mentioned case the training data can't be classified with the linear classifiers that we are already acquainted with. Thus came the idea of kernels which is a similarity function. It is a function that is provided to a machine learning algorithm which takes two inputs and shows how similar they are.

II. SVM MODEL

SVMs are machine learning tool that analyze data and recognize patterns or decision boundaries within the dataset used mainly for classification and regression analysis. SVM

constructs hyper-planes in a multidimensional space that separates different class boundaries and the number of dimensions is called the feature vector of the dataset. SVM has the capability to handle multiple continuous and categorical variables as shown in Fig.2 [5]. There are two kinds of circles, one filled and one outlined. The goal of the SVM is to separate the two types into classes based on the features. The model consists of three lines. One is w.x-b=0, that is the marginal line or margin. The lines w.x-b =1 and w.x-b =-1 represents the position of the closest data points of both the classes. The circles lying on the hyper-plane are called the support vectors. The filled circle in the other class is called an outlier. It is ignored to avoid over-fitting and hence obtaining a nearly perfect classification. The objective of the SVM is to maximize the perpendicular distance between the two edges of the hyper-plane to minimize the occurrence of generalization error. Since the hyper-plane depends on the number of support vectors, the generalization capacity increases with decreasing support vectors.

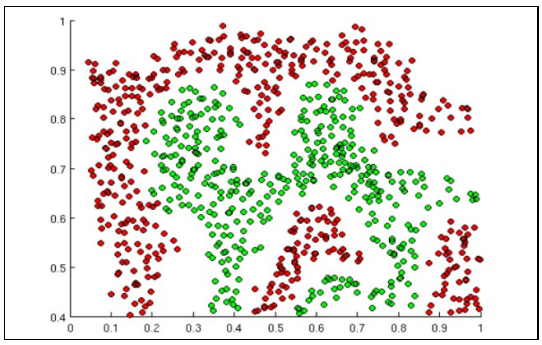

Fig. 1. Non-linear classification.

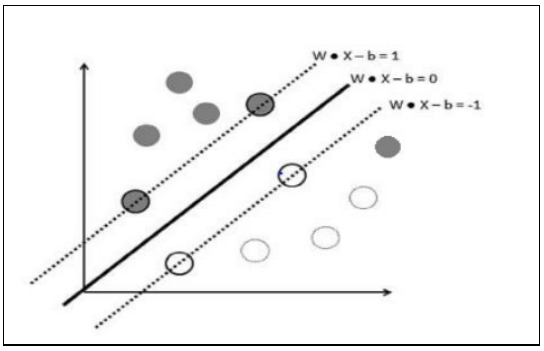

Fig. 2. SVM classifier.

A. Hyper-plane

SVM makes parallel partitions by generating two parallel lines to create a separate space in a high dimensional space using most of its attributes. This plane is called the hyper-plane. It creates hyper-planes that have the largest margin in the high dimensional space, hence separating the given data into classes and creating margins. The margin represents the maximum distance between the closest data points of the two classes. The larger the margin, the lower is the generalization error of the classifier. SVM provides the maximum flexibility of all the classifiers.

SVMs can be called as probabilistic approaches but do not consider dependencies among the attributes. SVM works on empirical risk minimization which leads us to an optimization function as in (3),

$$\min_{w} \sum l(x_i, y_i, w) + \lambda\, r(w) \qquad -(3)$$

Where l is the loss function (Hinge loss in SVMs) and r is the regularization function. SVM is a squared l_2-regularised linear model i.e. $r(w) = \| w_2 \|^2$. This bars us against large coefficients or coefficient magnitudes that are themselves penalized in the optimization. Regularization ($r(w)$) always gives us a unique solution in $m>n$ cases where m is the number of dimensions or features and n is the quantity of training data sets. Thus SVMs can be effective in a much higher dimensional space where the number of features in the feature vector are greater than number of training sample but may tend to get slow while learning as shown in Fig. 3 (a).

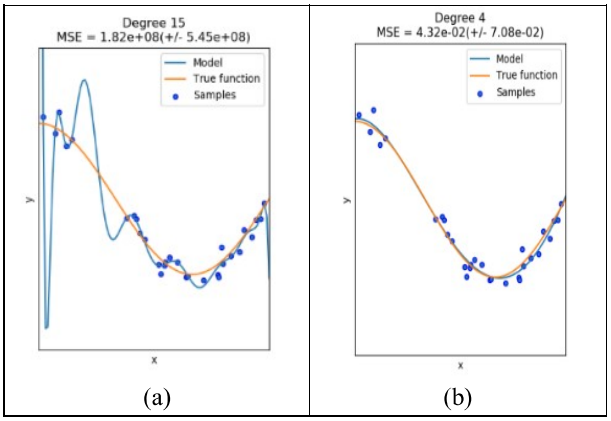

Fig. 3. (a) Over-fitting (b) Almost Perfect fit.

B. Over fitting

In the case of m>>n i.e. number of features is much greater than the number of training data samples, the regularization function introduces a bias so large that the training data model heavily underperforms causing over fitting as shown in Fig.3(b) [6]. Over fitting is one of the major curses of dimensionality. This phenomenon can be overcome by tuning the regularization parameter carefully in cases of linear regression and selecting the proper kernel and tuning them carefully.

C. Algorithm of SVM

There are two cases, separable case and non separable case, of which, Separable Case is where infinite boundaries are available to separate data into two classes. The boundaries giving the largest distance to the nearest observation is called the optimal hyper-plane. The optimal hyper-plane is derived using (4),

$$wx + b = 0 \qquad -(4)$$

This equation must satisfy the two conditions , one being it should separate the two classes A and B very well i.e. $f(x) = \omega.\, x + b$ is positive iff $x \in A$. And the other, it should exist further away from all the possible observations adding to the robustness of the model. Given that the distance from the hyper-plane to the observation x is $|\omega.x+b|/\|a\|$. The maximum margin should be $2/\|a\|$. Whereas in non separable

case the two classes cannot be separated properly, they overlap. A term measuring the error must add and the margins are normalized to $1/\|a\|$ giving a term i called the slack variable. Error of the model is the observation where $\xi > 1$ and sum of all the ξ_i gives the set of classification error. Thus, two constraints are formed to construct the hyper-plane, firstly for every i, $y_i.(\omega.x + b) >= 1 - \xi i$ and secondly $1/2\ \|a\|^2 + \delta\Sigma i\xi i$ is minimal. Quantity δ is parameter that penalizes errors. Increment of this increases the sensitivity of error and the adaptation of the model to the errors also increases.

III. Types of SVM Classifier

There are two kinds of classifier used for SVM such as linear and nonlinear which is discussed below.

A. Linear SVM

Here n training dataset(x1, y1 to xn, yn) are supplied. A large margin classifier between the two classes of data will be obtained. Any hyper-plane can be mentioned as the set of points $\vec{x}$ satisfying (5),

$$\vec{w}.\vec{x} - b = 0 \qquad -(5)$$

Where $\vec{w}$ serves as the normal vector to the hyper-plane. There can be of 2 types of margin, hard margin and soft margin. If the training data can be separable linearly and with completely without errors (outliers and noise), hard margin is used. In case of errors, either margin is smaller or hard margin fails. The hard margins are constructed in the following steps:

-It has to be enforced that all the points are out of margin i.e. $(w^T.\ x_j+b)\ y_j \geq a$

-The margin should be maximised i.e. Max $\gamma = a/\|\omega\|$ where a is the margin after points are projected onto ω.

- Finally, setting a to 1, we get min $\|\omega\|$ $(\omega^t.\ x_j + b) >=1$.

Dual form is when ω is a linear combination of training observations i.e. $\omega = \Sigma\alpha_l\ y_l\ x_l$ where α will be 0 except for support vectors. Soft margin can be called as extension of the hard margin. It is used in case of nonseparable classes i.e. overlapping classes as explained earlier. It introduces a slack variable ξi and ξj to calculate the net error in the classification i.e. min $\|\omega\| + C\ \Sigma\ \xi j\ (w^T.\ x_j+b)\ y_j >=1- \xi j$. In linear SVM two parallel hyper-planes are selected that distinguishes between the two classes of data, so that the distance between them can be maximized. Here by rescaling the datasets can be represented by the two equations for labeling each of the classes separated by the boundary using (6) and (7),

$$\vec{w}.\vec{x} - b = 1 \qquad -(6)$$

$$\vec{w}.\vec{x} - b = -1 \qquad -(7)$$

The classifiers are designed in such a way that anything above the boundary in (6) is of one class while anything below the mentioned constraint in (7) will be considered as other class.

B. Non-Linear SVM

This is the part where SVM plays its major role. Initially SVM was designed to serve for linear classifications. But later on in the late 20th century, it was designed to be used for non linear classification as well by the help of the kernels. The most common types of kernel are, polynomial kernel function, sigmoid kernel function and the RBF kernel function and a few more.

i. Polynomial kernel function

It works on SVM that represents the similarity of the training samples in a feature space over polynomials of the original variables. Polynomial kernel looks both at the given features of the input samples and the combination of them which are called as interaction features. In case of the boolean input features, the features are the logical conjunction of the given input. For degree *d* polynomials, the kernel works with the function in (7),

$$K(x,y)- (xy + C) \qquad -(7)$$

Where x and y are vectors in the input space and non negative C is a parameter which is meant to reduce the gap between the higher orders and the lower orders of the polynomial.ie, when C = 0, the polynomial is homogeneous. K is the inner product of the feature space based on a mapping φ i.e. $K(x,y) = <(\varphi(x), \varphi(y))>$. This kernel function finds its use in the natural language processing. In the Fig.4 shows how the kernel responds to the different degrees.

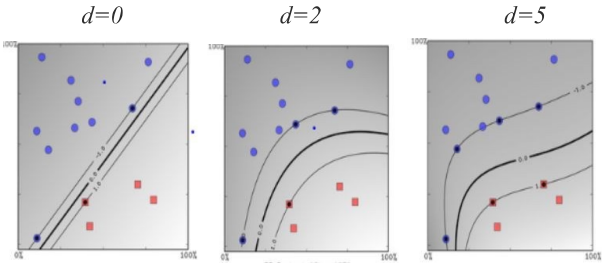

Fig. 4. Response of the kernel functions on different degrees.

ii. Sigmoid kernel function

The sigmoid kernel has its origin from the neural networks. It is generally not as effective as the RBF but it can be tuned to work approximately at par with the Gaussian RBF kernels. Problems where the number of feature vectors are high or non linear decision boundary in 2 dimensions, Sigmoid Kernels may be more or less as good as Gaussian RBF Kernels. The performance of the kernels in such situation depends upon the level of cross validation one needs to do for each. It finds it application as an activation function of artificial neurons as in equation (8),

$$K(x, y) = tanh(\propto x^T y + c) \qquad -(8)$$

Fig.5 shows how the surface plot of the C-SVM and the sigmoid function looks like.

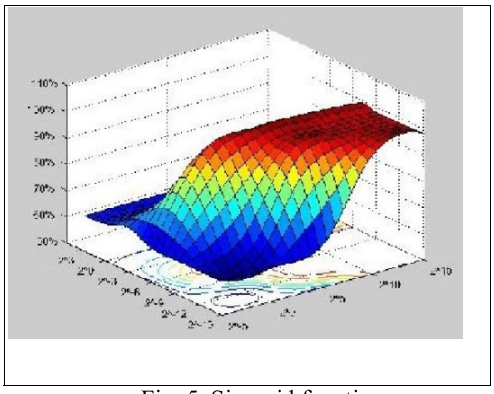

Fig. 5. Sigmoid function.

iii. Radial basis function kernel

It is a real valued function whose value depends on the Euclidean distance from the origin as given in (9)

$$\Phi(x, c) = \Phi(||x - c||) \qquad \text{- (9)}$$

Radial basis function kernel is basically a similarity function which calculates the Euclidean distance between two landmarks with a free parameter σ. In SVMs we use this radial functions to define the Gaussian radial basis function (RBF). This the most popular kernel used for classification of training data in SVM which given in (10),

$$K(x, x') = exp\,(-\gamma||x - x'||^2) \qquad \text{- (10)}$$

Further the parameter (γ) can also be written as in (11),

$$\gamma = \frac{1}{2\sigma^2} \qquad -(11)$$

where σ is a free parameter. When ($x \approx x'$) the value of the function is 1 while x is farthest from x', its value is 0 (in the limit) with respect with to standard value of σ2 =1, it can be interpreted readily a as a similarity measure.

Further if any change is made to the parameter, it must be noted that larger the value of σ^2, the features will vary more smoothly. It will have higher bias and lower variance. While for smaller value of σ^2 the features will vary less smoothly and will have lower bias and higher variance. RBF finds its application in detecting epileptiform artifacts in EEG recordings. A comparative plot is further shown in Fig.6.

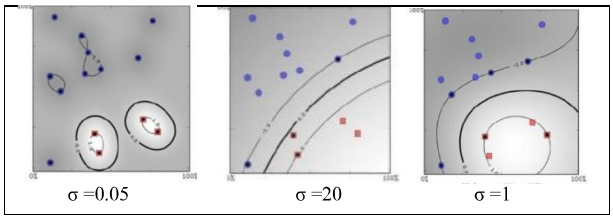

Fig. 6. Variation in the Gaussian RBF Kernel with variation in σ.

In order to get a comparative study of the plots of the different SVMs, the following output in Fig.7 is generated in spyder. The 1st column for Linear SVM, 2nd one for RBF, 3rd one for Polynomial, and the last one for sigmoid under different parameters.

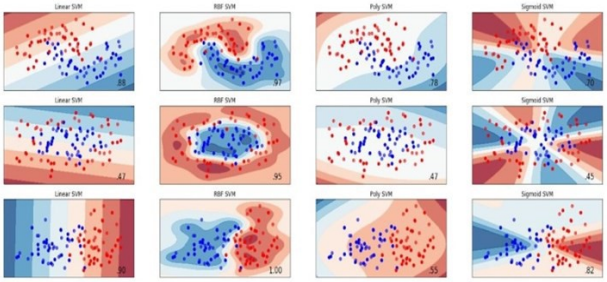

Fig. 7. A comparative study of the output of different kernel functions.

IV. Real Life Applications of SVM

A. Spam filtering

SVM finds its application in email spam filtering. In SVM library there are 4 type of kernels namely linear kernel, polynomial kernel, RBF) kernel and sigmoid kernel. After training the data set under these kernels, it has been found out that highest accuracy is obtained at 92.4381% using the c-svc parameter for linear kernel.

B. Facial detection

SVM is used in face detection techniques. One reasons for the applications of SVM is that its decision function only requires the dot product of the feature vector with the Support Vector machine , i.e. knowledge about its dimension is not necessary which is why it produces best results.

C. Text categorization

Text categorization finds its importance in information retrieval. It classifies documents into a set of predefined categories.In a comparative study using Naive Bayes, SVM and Neural Network, it has been found out that SVM has generated the best results.

D. Bioinformatics

SVM find its application in bioinformatics.For example it can be used for prediction of the extended early-stage (ES) protein structures by designing a classifier. SVM shows satisfactory results on several biological pattern classification problems which is why it has become a standard tool in bioinformatics.

E. Predictive control of environmental disasters

SVM finds its applications in prediction of Natural calamities like earthquakes and tsunamis. The processed signal of the seismic wave is fed to the SVM classifier as input which further predicts the location of the forthcoming earthquakes. It has been noted that regarding the location of future earthquakes it could give an accuracy of 77%.

V. Conclusion and Future Work

In this paper concept of SVM along with its various models has been presented. Phenomenon which leads to the concept of SVM back in the 90s' is mentioned. SVMs are mainly used for classification and regression analysis both for linear and non linear decision boundaries. SVM classify non linear decision boundaries with the help of kernels namely RBF, sigmoid, polynomial etc. Barring a few limitations SVM, it is a very useful machine learning tool which finds its application in our day to day life such as spam filtering, facial detection etc. It gives the most optimized and accurate results in its domain when compared

with other algorithms. A classifier is judged on the basis of training time, testing time, classification accuracy. It will be very challenging to determine a relationship between the kernels and its distribution data which will choose a proper kernel function for a given dataset in order to maximize class separability between data points. It can also been seen that nowadays mobile applications collect huge amount of user data for better in-app experience of the users. This is can be done using properly scalable kernels, using which large volumes of data collection won't be necessary.

REFERENCES

[1] V. Vapnik, "Statistical Learning Theory", Wiley-Interscience, Publication, New York, 1998.

[2] V.N. Vapnik, "Principles of Risk Minimization for Learning Theory". In Proceedings of Advances in Neural Information Processing Systems, 1992.

[3] C. Junli, and J. Licheng, "Classification mechanism of support vector machines," 5th International Conference on Signal Processing Proceedings. 16th World Computer Congress, Beijing, 2000, vol.3, pp. 1556-1559, 2000.

[4] L. Rosasco, E.D. De Vito, A. Caponnetto, M. Piana, A. Verri, "Are Loss Functions All the Same?" Neural Computation, vol. 16, 2004.

[5] C.J.C Burges, "A Tutorial On Support Vector Machines for Pattern Recognition", Data Mining and Knowledge Discovery , vol. 2, no. 2, pp. 121-167, 1998.

[6] H. Han, J. Xiaoqian, "Overcome Support Vector Machine Diagnosis Overfitting." Cancer Informatics, vol. 13, suppl 1, pp. 145–158, 2014.

[7] N. Gruzling, "Linear separability of the vertices of an n-dimensional hypercube", M.Sc Thesis, University of Northern British Columbia, 2006.

[8] D. Chaudhuri, and B.B. Chaudhuri, "A novel multiseed nonhierarchical data clustering technique," IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 27, no. 5, pp. 871-876, Sep 1997.

[9] S.P. Boyd, L. Vandenberghe, "Convex Optimization" Cambridge University Press. ISBN 978-0-521-83378-3, 2004.

[10] M.A. Aizerman, E.M. Braverman, L.I. Rozonoer, "Theoretical foundations of the potential function method in pattern recognition learning", Automation and Remote Control, vol. 25, pp. 821–837, 1964.

[11] B.E. Boser, I.M. Guyon, V.N Vapnik, "A training algorithm for optimal margin classifiers". Proceedings of the fifth annual workshop on Computational learning theory, pp. 144, 1992.

[12] Y. Goldberg and M. Elhadad, "SVM: Fast, Space-Efficient, non-Heuristic, Polynomial Kernel Computation for NLP Applications", Proc. ACL-08: HLT, 2008.

[13] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee, "Choosing multiple parameters for support vector machines", Machine Learning, vol. 46, pp. 131–159, 2002.

[14] A. Shashua, "Introduction to Machine Learning: Class Notes 67577". arXiv:0904.3664v1 Freely accessible [cs.LG], 2009.

[15] C. Cortes, V. Vapnik, "Support-vector networks", Machine learning, vol. 20, no. 3, pp. 273-297, 1995.

[16] Y.W. Chang, C.J. Hsieh, K.W. Chang, M. Ringgaard, C.J. Lin, "Training and testing low-degree polynomial data mappings via linear SVM", Journal of Machine Learning Research, Vol. 11, pp. 1471–1490, 2010.

[17] A. Ben-Hur, C.S. Ong, S. Sonnenburg, B. Schölkopf, G. Rätsch, "Support Vector Machines and Kernels for Computational Biology", PLoS computational biology, vol. 4. 2008.

[18] D. Michie, D.J. Spiegelhalter, and C.C. Taylor, "Machine Learning, Neural and Statistical Classification", Englewood Cliffs, N.J.: Prentice Hall, 1994.

[19] J. Vert, K. Tsuda, and B. Schölkopf, "A primer on kernel methods", 2004.

[20] D.S. Broomhead, D. Lowe, "Multivariable functional interpolation and adaptive networks", Complex Systems, vol. 2, pp. 21-355, 1988.

[21] D.K. Agarwal and R. Kumar, "Spam Filtering using SVM with different Kernel Functions", International Journal of Computer Applications, Vol. 136, No. 5, February 2016.

[22] Irene Kotsia, Nikolaos Nikolaidis, and Ioannis Pitas, "Facial expression recognition in videos using a novel multi-class support vector machines variant", Aristotle University of Thessaloniki, Department of Informatics, Thessaloniki, Greece.

[23] S. Alsaleem, "Automated Arabic Text Categorization Using SVM and NB", International Arab Journal of e-Technology, Vol. 2, No. 2, 2011.

[24] P. Fabian and K. Stąpor, "Developing a new SVM classifier for the extended ES protein structure prediction," Federated Conference on Computer Science and Information Systems, Prague, pp. 169-172, 2017.

[25] W. Astuti, R. Akmeliawati, W. Sediono, M.J.E. Salami, "Hybrid Technique Using Singular Value Decomposition (SVD) and Support Vector Machine (SVM) Approach for Earthquake Prediction", IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, Vol. 7, Issue. 5, pp. 1719 – 1728, 2014.