

Performance Analysis of ANN on Dataset Allocations for Pattern Recognition of Bivariate Process

Olatunde .A. Adeoti^{1*} Peter .A. Osanaiye²

1. Department of Mathematics and Statistics, Bowen University, Iwo, Osun state, Nigeria.
2. Department of Statistics, University of Ilorin, Ilorin, Nigeria.

*Email of the corresponding author: tim_deot@yahoo.com

Abstract

Several approaches to identifying the out-of-control variables after the detection of abnormal pattern has been most intensively studied and used in practice. One of the several approaches is the Artificial Neural Network (ANN) based model for diagnosis of out-of-control signal of multivariate process mean shift. In spite of the number of years of research in neural network, limited research (if any) have been done on the effect of dataset allocations in percentages for training and testing on the performance of ANN. In this paper, we investigate the use of different percentages of dataset allocation into training, validation and testing on the performance of ANN in pattern recognition of bivariate process using six selected training algorithms. The result of study showed that large allocation of dataset for training was found suitable, having higher recognition accuracy for ANN learning and perform better for pattern recognition of bivariate process.

Keywords: Bivariate Process; Pattern Recognition; Recognition accuracy; Multivariate quality control charts, training algorithm

1. Introduction

The control charts are powerful tools in SPC for the study and control of repetitive process as well as detecting an out-of-control situation. Control chart is of univariate and multivariate type. Univariate chart is used to monitor processes that manufacture products with a single quality characteristic. An obvious advantage of the chart is that the interpretation of a signal is straight forward since the source of the problem arises from a shift in the mean or variance of the quality variable being monitored. Multivariate chart on other hand is used to monitor processes that manufacture products with two or more quality variables that are usually correlated. In monitoring the multivariate process and determining if it is in control, several multivariate charts are used. They include Hotelling T^2 (Hotelling, 1947), MEWMA (Lowry et al., 1992; Pignatiello and Runger, 1990) and MCUSUM (Woodall and Ncube, 1985; Healy, 1987; Crosier, 1988). A recent review of multivariate statistical process control was given by Bersimis et al. (2007). Though the multivariate charts can monitor multivariate process, the problem of these charts is the

difficulties in detecting which variable or set of variables that is responsible for the signal when the process is out of control. Both analytical and graphical methods have been proposed to overcome this problem with associated deficiency. (See Alt, 1985; Jackson, 1985; Murphy, 1987; Doganaksoy et al., 1991; Mason et al., 1995, 1997; Fuchs and Benjamini, 1994)

Advanced computing technologies and data-collection systems has motivated many researchers to explore the use of Artificial Neural Networks (ANNs) based model (Rumelhart et al. 1986; Anagun, 1998; Chen and Wang, 2004; Guh and Hsieh, 1999). However, one of the most important problems that neural network designers face is an appropriate network size as well as appropriate percentages of dataset allocation into training, validation and testing for a given application. Network size involved the case of layered neural network architectures, the number of layers in a network, the number of nodes per layer and the number of connections. Haykin (1999) stated that while it is possible to design an artificial neural network with no hidden layer, they can only classify data that is linearly separable which severely limits their applications. Medsker and Liebowitz (1994) opined that ANN that contains hidden layers have the ability to deal robustly with nonlinear and complex problems and thus can operate on more interesting problems. On the percentages of dataset allocation for training ANN, different percentages of dataset allocation have been suggested in the literature but there are no mathematical rules for the determination of the required sizes of the various subsets. Looney (1996) recommends 65% of the dataset to be used for training, 25% for testing and 10% for validation (cited in Basheer and Hajmeer, 2000), whereas Demuth et al., (1998) proposed 60% for training, 20% for validation and 20% for testing same as Niaki and Abbasi (2005). In this study, we examined the performance of ANN on different percentages of dataset allocation into training, validation and testing levels using six different training algorithms.

The paper is divided into five sections. Section 2 consists of fundamentals of neural network including network configuration, dataset allocation and training algorithms. Section 3 is the experimental procedure. Section 4 is the discussion of the results and section 5 gives the conclusion.

2 Fundamentals of Artificial Neural Network (ANN)

Artificial Neural Networks (ANN) are computational modeling tools that have recently emerged and found extensive acceptance in many disciplines for modeling complex real world problems. ANN is an adaptive, most often nonlinear system which consists of a set of computational units called neurons or nodes and a set of weighted directed connections between these nodes. Adaptive means that the network learns by examples as the parameters are changed during operation which is normally called training phase. Schalkoff (1997) define neural network as structures comprised of densely interconnected adaptive simple processing elements (called artificial neurons or nodes) that are capable of performing massively parallel computations for data processing and knowledge representation. ANN has the ability to learn complex nonlinear input/output relationships, use sequential training procedure, adapt themselves to the data and generalize knowledge.

The Artificial Neural Network is built with a systematic step-by-step procedure to optimize a performance criterion or to follow some implicit internal constraint, which is commonly referred to as the learning rule. In artificial neural network, the designer chooses the network topology; the performance function, the learning rule and the criterion to stop the training phase but the system automatically adjust the parameters.

The increasing popularity of neural network models to solve complex real world problems has been primarily due to the availability of efficient learning algorithms for practitioners to use. ANN had been utilized in variety of field such as manufacturing process, financial, medical, telecommunications e.t.c.

2.1 ANN configuration

The most commonly used family of ANN for pattern recognition is the multilayer perceptron (MLP) network (Rumelhart et al., 1986; Sagrigolu et al, 2000). It consists of three layers: input, hidden and output layers. The input layer with nodes representing input variables to the problem, an output layer with nodes representing the dependent variables, that is, what is being modeled, and one or more hidden layers containing nodes to help capture the nonlinearity in the data. The primary function of the input layer is to distribute the input information to the next processing layer and the task of the output layer is to determine the pattern (Guh and Hsieh, 1999). The ANN model is trained and tested before it can be deployed for pattern recognition. Supervised training approach is adopted. Here, sets of data comprising input and target vectors are presented to the MLP network. The learning process takes place through adjustment of weight connections between the input and the hidden layer and between the hidden and output

layer to minimize error between the actual and desired output. The input node size is equal to the number of variables, i.e 2. The number of output nodes in the study is set corresponding to the number of pattern classes, i.e. 3. The number of nodes in the hidden layer is selected by varying the number of nodes between 10 and 20. The transfer functions used are hyperbolic tangent function for the hidden layer that transforms the layer input to output range from -1 to +1 and is defined as

$$f(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \quad (1)$$

and sigmoid function for the output layer that transforms the layer input to output range from 0.0 to 1.0 and is defined as

$$f(z) = \frac{1}{1 + e^{-z}} \quad (2)$$

2.2 Training Algorithms

Various types of training algorithms for which the multilayer perceptron (MLP) network learn exist but it is very difficult to know which training algorithm will be suitable for training ANN model for pattern recognition of bivariate manufacturing process. This depends on many factors including the number of weights and biases, error goal and number of training iterations (epochs). Backpropagation algorithm, the common and most widely used algorithm in training artificial neural network learns by calculating an error between desired and actual output and propagate the error information back to each node in the network. This propagated error is used to drive the learning at each node. There are many variations of the backpropagation algorithm. In this study, six training algorithms are evaluated for the different dataset allocations into training, validation and testing. They are: gradient descent algorithm, gradient descent with momentum, Levenberg-Marquardt, scaled conjugate, resilient backpropagation and quasi-Newton.

The Levenberg-Marquardt (trainlm) algorithm blends the steepest descent method and the Gauss-Newton algorithm as it inherits the speed of the Gauss-Newton and stability of the steepest descent method. It provides a numerical solution to the problem of minimizing a function, generally nonlinear, over a space of parameters of the function. It is fast and has stable convergence and it is able to obtain lower mean square error than any other algorithms. The scaled conjugate gradient algorithm (trainscg) was designed to avoid the time-consuming line search of the conjugate gradient algorithms which is computationally expensive, because it requires that the network response to all training inputs be computed several times for each search. The trainscg is fast as trainlm and able to obtain lower

mean square. Quasi-Newton (trainbfg) is a class of algorithms that is based on Newton's method which doesn't require calculation of Hessian matrix of second derivatives but consider an approximation of the Hessian matrix specified by gradient descent that is suitably updated at each iteration of the algorithm. The weights and biases of gradient descent (traingd) are updated in the direction of the negative gradient of the performance function and only after the entire training set have been applied to the network. The gradient descent with momentum (traingdm) allows a network to respond not only to the local gradient, but also to recent trends in the error surface. The gradient descent is usually slower in convergence. The resilient backpropagation (trainrp) is a first order optimization technique for minimizing the error function. The purpose of the resilient backpropagation (Rprop) training algorithm is to eliminate the harmful effects of small magnitudes of the partial derivatives when steepest descent is used to train a multilayer network with sigmoid functions since sigmoid functions are characterized by the fact that their slopes must approach zero as the input gets large. The memory requirement for Rprop is relatively small compared to other algorithms. The ANN model is trained with these algorithms and the experiment is coded in MATLAB using its ANN toolbox (Demuth et al 2010)

3. Experimental procedure

ANN model was developed using raw dataset as the input vector. This section discusses the neural network design, procedure for data generation and allocation of dataset into training, validation and testing of the model

3.1 Dataset generation and Allocation

Ideally, the best way to obtain dataset for training and testing of ANN model is to collect various data from real-world manufacturing process environment. Since the real data in the production process are not economically available and difficult to obtain, simulation is an effective and useful alternative generating process data. This study simulates the data from bivariate normal distribution when the variables are correlated. Let $X \sim N_2(\mu, \Sigma)$ denote the bivariate normal distribution with mean μ and variance-covariance matrix $\Sigma = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$ where $\sigma_{i,i}=1$, for all i and $\sigma_{i,j}=\rho$ for all $i \neq j$ and ρ is the correlation value between the two variables in each pattern. The in-control mean is assumed to be a zero vector. The variance-covariance matrix is assumed to be scaled so as to have unit variance for all components. The dataset had 150 examples from each class (i.e. 450 examples). After the data generation step, an important step is the allocation of the dataset into training, validation and testing data. The

training dataset is used for computing gradient and updating the weight connections among the layers. Validation set is used to monitor training process to avoid overfitting and bias while the testing set is used for preliminary performance test of generalization to data not yet seen. Chen and Wang (2004) randomly allocated the dataset of seventy-five input vectors for nine distinct types of shift which was generated by monte-carlo method into training and testing in the ratio of 70:30. Niaki and Abbasi (2005) proposed a neural network for fault diagnosis of bivariate process and use large dataset of five hundred for each pattern cases. Demuth et al. (2010) proposed the allocation of dataset into 60% (training), 20% (validation) and 20% (testing) which was adopted by Kiran et al (2010) before they were presented to the ANN for the learning process. In this study, the allocation of dataset were randomly allocated into 50% (training), 25% (validation) and 25% (testing); 60% (training), 20% (validation) and 20% (testing); 70% (training), 15% (validation) and 15% (testing) and 80% (training), 10% (validation) and 10% (testing) for the ANN learning process.

3.2 Neural Network training

The dataset were randomly allocated using the different percentages of dataset allocation of the study to avoid possible bias in the presentation order. The network was trained using the different dataset allocation into training (in percentages) for updating the network weights and the validation set for in-training validation. Six training algorithms were employed in the training of the model for the different dataset allocations. The number of nodes in the hidden layer was varied between 10 and 20. The training process was stopped whenever the maximum number of validation failures was exceeded or the maximum allowable number of epochs was reached or the error goal was achieved. The maximum allowable number of epoch was 5000 and the performance error goal was set at 10^{-8} . MATLAB M-files were developed for the training and diagnosis performance of the network using the MATLAB Neural network toolbox software. The trained ANN model was evaluated to obtain the recognition accuracy once the training stopped. The mean square error for the dataset allocation were observed and the dataset allocation that gives the minimum mean square error and maximum recognition accuracy for the ANN model was considered the best for pattern recognition of bivariate process.

4 Results and Discussion

The result in Table 1 and figure 2 show that the recognition accuracy of the pattern recognition neural network improves marginally as percentages of dataset allocated into training set also increases for Levenberg-Marquardt algorithm, Quasi-Newton and Resilient backpropagation but deteriorates for other algorithms. For three of the dataset allocations where the training subset is smaller than 80%, the performance of the ANN is mostly affected by the Quasi-Newton and the remaining allocation affected by Levenberg-Marquardt algorithm. This demonstrates that when more of the dataset is allocated for training, the recognition performance of the neural network model also improves. A MATLAB output of the training of one of the ANN model is shown in Figure 1. The best recognition accuracy (the highest value) occurred when dataset are allocated into 70% (Training), 15% (Validation) and trained with Quasi-Newton algorithm and when dataset are allocated into 80% (Training), 10% (Validation) and trained with Levenberg-Marquardt algorithm, while its generalization performance is only being assessed based on the remaining percentages of the testing subset. Therefore, it is recommended that larger percentages of the dataset be allocated for the training of neural network for optimal performance of the pattern recognition neural network of bivariate process control.

Similarly, the network trained with the Levenberg-Marquardt algorithm gives the lowest mean square error when compared to other algorithms for the different percentages of dataset allocations in table 2

The effect of random allocation of dataset into training, validation and testing cannot be neglected in the optimal performance of the bivariate neural network. Overall, the performance of ANN improves when larger percentages of the dataset are allocated for training (learning) of the sample data and trained with Quasi-Newton or Levenberg-Marquardt algorithm. However, Levenberg-Marquardt seems to be better because of smaller mean square error performance.

5. Conclusion

This paper examine the effect of different percentages of dataset allocation into training, validation and testing on the artificial neural network with the aim of obtaining a suitable dataset allocation in percentages for training, validating and testing of ANN model. The MLP neural network was utilized because studies show that it has better performance for pattern recognition. Four different groupings of dataset allocation namely: 50% (Training), 25% (Validation) and 25% (Testing); 60% (Training), 20% (Validation) and 20% (Testing); 70% (Training), 15% (Validation) and 15% (Testing); 80% (Training), 10% (Validation) and 10% (Testing) were evaluated with six

training algorithms and the dataset allocation into 80% (Training), 10% (Validation) and 10% (Testing) trained with Levenberg-Marquardt algorithm is identified to be the best allocation for the problem because it has good recognition accuracy and minimum mean square error compared to other dataset allocation.

References

- Alt, F.B. (1985) "Multivariate quality control", The Encyclopedia of Statistical Sciences Kotz S, Johnson NL, Read CR (eds). Wiley: New York, 110-122
- Anagun, A.S. (1998) "A neural network applied to pattern Recognition in Statistical Process Control". *Computers and Industrial Engineering* **35**, 110-122
- Basheer, I. A and Hajmeer, M (2000) "Artificial neural networks: fundamentals, computing, design, and application". *Journal of Microbiological Methods* **43**, 3–31
- Bersimis, S., Psarakis, S. and Panaretos, J. (2007) "Multivariate Statistical Process Control Charts: An overview". *Quality and Reliability Engineering International* **23**, 517-543
- Chen, L.H. and Wang, T.Y (2004). "Artificial Neural Networks to classify mean shifts from multivariate χ^2 chart signals". *Computers and Industrial Engineering* **47**, 195-205.
- Crosier, R.B. (1988) "Multivariate generalizations of cumulative sum quality-control schemes", *Technometrics* **30**, 291-303
- Demuth H, Beale M and Hagan A (2010) Neural Network Toolbox User's Guide. Math Works, Natick
- Doganaksoy, N., Faltin, F.W., and Tucker, W.T. (1991) "Identification of out-of-control multivariate characteristic in a multivariable manufacturing environment" *Communication in Statistics: Theory and Methods* **20**, 2775–2790
- Fuchs, C. and Benjamini, Y. (1994) "Multivariate profile charts for statistical process control" *Technometrics* **36**, 182-195
- Guh, R.S. and Hsieh, Y.C. (1999). "A Neural Network Based Model for Abnormal Pattern Recognition of Control Charts", *Computers and Industrial Engineering* **36**, 97-108
- Haykin, S (1999). Neural Networks: A comprehensive Foundation. Second edition, Prentice-Hall, New Jersey
- Healy, J.D. (1987) "A note on multivariate CUSUM procedures", *Technometrics* **29**, 409- 412
- Hotelling, H. (1947) "Multivariate Quality Control-Illustrated by the Air Testing of Sample Bombsights," *Techniques of Statistical Analysis* (Eisenhart, C., Hastay, M. W., and Wallis, W. A. eds.), McGraw Hill, New York
- Indra Kiran, N.V.N, Pramila Devi, M and Vijaya Lakshmi, G (2010) "Effective Control Chart Pattern Recognition Using Artificial Neural Network". *International Journal of Computer Science and Network Security*, **10**(3), 194-199
- Lowry, C.A, Woodall W.H., Champ C.W. and Rigdon S.E. (1992) "A multivariate EWMA control chart" *Technometrics* **34**, 46-53
- Mason, R. L., Tracy, N. D. and Young, John C. (1995) "Decomposition of T^2 for Multivariate Control Chart Interpretation," *Journal of Quality Technology*, **27**(2), 99-109.
- Mason, R. L., Tracy, N. D. and Young, J.C. (1997), "A Practical Approach for Interpreting Multivariate T^2 Control Chart Signals". *Journal of Quality Technology*, **29**(4), 396-406.
- Medsker, L and Liebowitz, J. (1994) Design and development of Expert systems and neural networks, New York.
- Murphy, B. J. (1987) "Selecting out of control variables with the T^2 multivariate quality control procedure". *The Statistician*, **36**, 571–583

- Niaki, S.T.A and Abbasi, B. (2005) "Fault diagnosis in multivariate control chart using artificial neural networks". *Quality Reliability Engineering International* 21, 825–840
- Pignatiello J.J and Runger G.C. (1990) "Comparisons of multivariate CUSUM charts" *Journal of Quality Technology* 22, 173-186
- Rumelhart, D. E., Hinton, D. E., and Williams, R. J. (1986), "Learning Internal Representations by Error Propagation in Parallel Distributed Process" MIT Press, Cambridge, MA, 318-362.
- Sagiroglu, S., Besdok, E. and Ertler, M (2000) "Control chart Pattern Recognition Using Artificial Neural Networks". *Turkish Journal of Electrical Engineering*. 8(2), 137-147.
- Schalkoff, R.J. (1997). *Artificial Neural Networks*. McGraw-Hill: New York
- Woodall W.H. and Ncube M.M (1985) "Multivariate CUSUM quality control procedures". *Technometrics* 27, 285-292

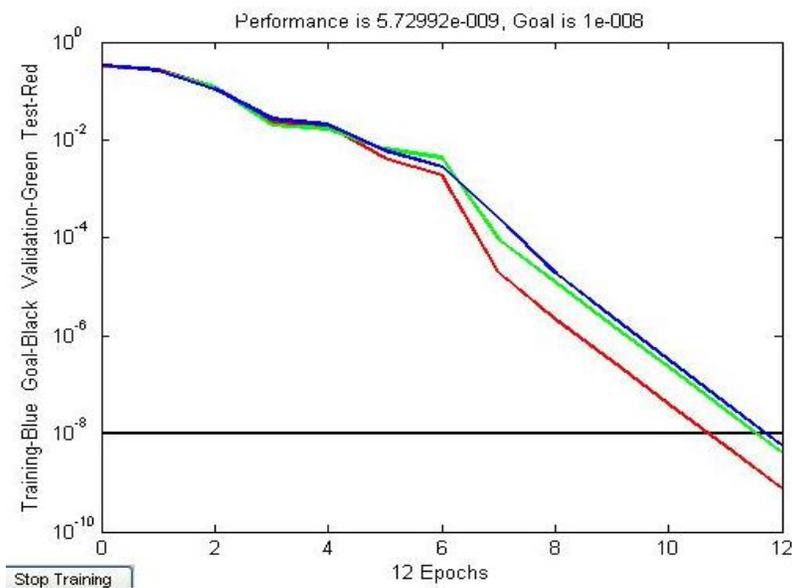


Figure 1 A MATLAB output of ANN training with dataset allocation into training, validation and testing

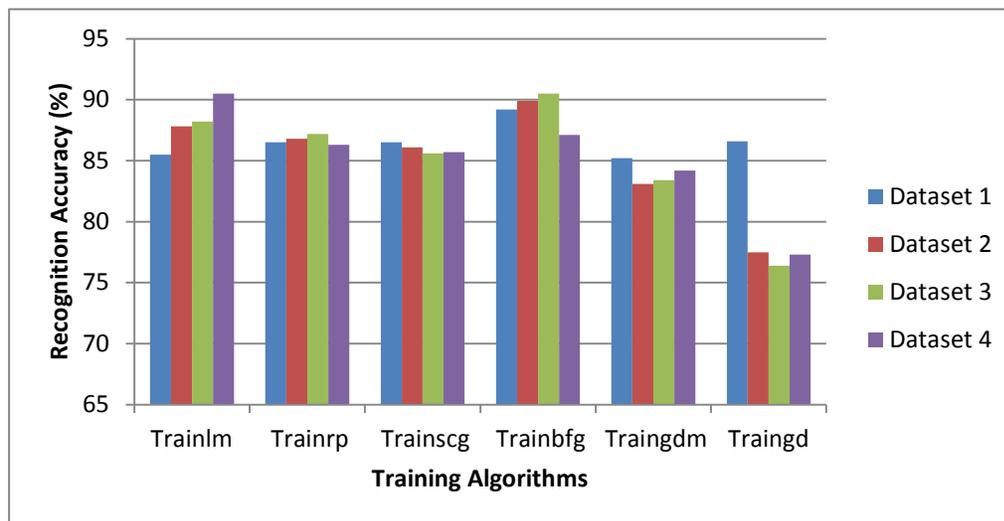


Figure 2 Performance of ANN for different percentages of dataset allocation into training, validation and testing where dataset 1 is 50% (Training), 25% (Validation) and 25% (Testing), dataset 2 is 60% (Training), 20% (Validation) and 20% (Testing), dataset 3 is 70% (Training), 15% (Validation) and 15% (Testing) and dataset 4 is 80% (Training), 10% (Validation) and 10% (Testing)

Table 1

Recognition Accuracy of trained ANN on percentages Dataset Allocation with six different algorithms

Training Algorithms	Percentages of Dataset Allocations			
	50% (Training), 25% (Validation) and 25% (Testing)	60% (Training), 20% (Validation) and 20% (Testing)	70% (Training), 15% (Validation) and 15% (Testing)	80% (Training), 10% (Validation) and 10% (Testing)
Levenberg-Marquardt (Trainlm)	85.5	87.8	88.2	90.5
Resilient Backpropagation (Trainrp)	86.5	86.8	87.2	86.3
Scaled Conjugate Gradient (Trainscg)	86.5	86.1	85.6	85.7
Quasi-Newton (Trainbfg)	89.2	89.9	90.5	87.1
Gradient Descent with Momentum (Traingdm)	85.2	83.1	83.4	84.2
Gradient Descent (Traingd)	86.6	77.5	76.4	77.3

Table 2

Mean square error performance of ANN model on percentages Dataset Allocation with six different algorithms

Training Algorithms	Percentages of Dataset Allocations			
	50% (Training), 25% (Validation) and 25% (Testing)	60% (Training), 20% (Validation) and 20% (Testing)	70% (Training), 15% (Validation) and 15% (Testing)	80% (Training), 10% (Validation) and 10% (Testing)
Levenberg-Marquardt (Trainlm)	1.04036×10^{-13}	1.53983×10^{-9}	4.316×10^{-14}	4.71267×10^{-27}
Resilient Backpropagation (Trainrp)	5.84036×10^{-9}	8.76111×10^{-9}	1.36228×10^{-8}	1.90931×10^{-10}
Scaled Conjugate Gradient (Trainscg)	3.20737×10^{-11}	1.84296×10^{-7}	1.83487×10^{-8}	2.20146×10^{-7}
Quasi-Newton (Trainbfg)	1.78209×10^{-8}	2.66866×10^{-8}	1.29354×10^{-9}	4.28321×10^{-8}
Gradient Descent with Momentum (Traingdm)	0.0103566	0.00788974	0.0343137	0.0241588
Gradient Descent (Traingd)	0.0316514	0.113465	0.0418345	0.00982071

This academic article was published by The International Institute for Science, Technology and Education (IISTE). The IISTE is a pioneer in the Open Access Publishing service based in the U.S. and Europe. The aim of the institute is Accelerating Global Knowledge Sharing.

More information about the publisher can be found in the IISTE's homepage:

<http://www.iiste.org>

CALL FOR PAPERS

The IISTE is currently hosting more than 30 peer-reviewed academic journals and collaborating with academic institutions around the world. There's no deadline for submission. **Prospective authors of IISTE journals can find the submission instruction on the following page:** <http://www.iiste.org/Journals/>

The IISTE editorial team promises to review and publish all the qualified submissions in a **fast** manner. All the journals articles are available online to the readers all over the world without financial, legal, or technical barriers other than those inseparable from gaining access to the internet itself. Printed version of the journals is also available upon request of readers and authors.

IISTE Knowledge Sharing Partners

EBSCO, Index Copernicus, Ulrich's Periodicals Directory, JournalTOCS, PKP Open Archives Harvester, Bielefeld Academic Search Engine, Elektronische Zeitschriftenbibliothek EZB, Open J-Gate, OCLC WorldCat, Universe Digital Library, NewJour, Google Scholar

