Journal of Statistics, Computing and Interdisciplinary Research
ISSN (P): 2707-7101

J. Stat. Comp. & Interdis. Rese.
Volume 3(1) (2021), 57-72
http://scir.wum.edu.pk/index.php/ojs

Generalization ability of Extreme Learning Machine using different Sample Selection Methods

Saher Fatima[1], Rana Aamir Raza[1*], Maruf Pasha[2], Asghar Ali[3] and Saba Kanwal[1]

[1]Department of Computer Science, Bahauddin Zakariya University, Multan, Pakistan.
[2]Department of Information Technology, Bahauddin Zakariya University, Multan, Pakistan.
[3]The Women University, Multan, Pakistan.

[*]Corresponding Author: aamir@bzu.edu.pk

Abstract

. The recent explosion of data has triggered the need of data reduction for completing the effective data mining task in the process of knowledge discovery in databases (KDD). The process of instance selection (IS) plays a significant role for data reduction by eliminating the redundant, noisy, unreliable and irrelevant instances, which, in-turn reduces the computational resources, and helps to increase the capabilities and generalization abilities of the learning models. . This manuscript expounds the concept and functionalities of seven different instance selection techniques (i.e., ENN, AllKNN, MENN, ENNTh, Mul- tiEdit, NCNEdit, and RNG), and also evaluates their effectiveness by using single layer feed-forward neural network (SLFN), which is trained with extreme learning machine (ELM). Unlike traditional neural network, ELM randomly chooses the weights and biases of hidden layer nodes and analytically determines the weights of output layer node. The generalization ability of ELM is analyzed by using both original and reduced datasets. Experiment results depict that ELM provides better generalization with these IS methods

Keywords: Data reduction, extreme learning machine, neural network.

1. Introduction

Learning from examples is one of the most important paradigms in the data mining and machine learning. According to [1], the problem of learning from the data (i.e., a classification process) can be formulated as: For a dataset D, a hypothesis H, and a performance measure P, the learning model L outputs a hypothesis h ∈ H, which optimizes P. D consists of N training examples (i.e., D = {1 • • • N}) which are called instances. Each instance consists of set of attributes, one of which is an output called dependent variable or class variable and remaining are independent variables or inputs also called features. h is called a classifier (i.e., learning model) which is based on D. The process of data reduction is considered as an important task in the knowledge discovery in databases (KDD).

Authors in [2] mentioned that KDD refers to the nontrivial process of identifying potential and useful patterns in data. KDD assumes that data reduction plays an essential role in successful data

mining process and it cannot be ignored.

From the literatures, one can see that much of the research works from the KDD domain, focus on either by scaling up machine learning algorithms or scaling down the data [3]. Instance selection is one of the data reduction techniques [4] and having many advantages such as it helps to increase capabilities and generalization performance of the classification model [5], reduces the space complexity [6], decreases the computational time and speeds up the knowledge extraction process [3, 7, 8].

In a classification process, Machine learning deals with algorithms that allow machinesto generate trained models after learning from data. Supervised learning algorithm analyzes labeled data and then produces a model that is used for mapping unseen inputs to the outputs. Results produced by supervised learning are more accurate than unsupervisedlearning that labels the unknown inputs by finding the patterns in the data. Training of machine learning algorithms over large datasets requires high computational power (i.e., in term of run-time and memory consumption) which is attracting researcher's concern towards the reduction of datasets. Most widely used data reduction techniques are: feature reduction (reducing the number of columns of a dataset) and instance reduction (reducingthe number of rows of a dataset). Both the processes are shown in the figure below:

Instance selection shrinks the dataset by picking useful instances from original datasets (as depicted in Figure 1) and increases the capabilities of the classifiers and their run-time by removing the redundant and incorrect entries [9].

Figure 1: Instance selection process

For a given dataset *D*, instance selection is the practice of selecting fewer instances from *D* and adding it to another dataset (i.e reduced dataset) R by leaving the noisy and duplicate instances in D. So, we can say that *R* is the subset of *D* where it can be assumed that accuracy improves for R, (i.e., $R_{acc} \geq D_{acc}$).

Note that even if the accuracy does not increase for R, other goals will be accomplished that includes the reduction in storage requirements of data set and reduction in training time of classifier and lastly reduction in total time taken to label the unknown inputs. Different instance selection methods are discussed in literature. They may be incremental, decremental or batch [10]. In incremental methods, *R* is initially empty and according to specified criteria, instances from *D* are selected and added to *R*. In decremental methods, *R* is initially equal to *D* and instances are removed from it one by one according to specified criteria. Lastly, in methods that use batch approaches, all instances that do not qualify specified criteria are removed at once from *D* and remaining instances becomes the part of *R*.

In this paper, different instance selection techniques are discussed and evaluated to analyze their impact on the generalization performance of extreme learning machine (ELM). Rest of the paper is divided as follows: Section 2 provides the detail prologue about instance selection techniques that are used with learning model. In section 3, an SLFN called Extreme Learning Machine (ELM) is presented. Experiments are performed in section 4. Results are discussed in section 5 and finally, section 6 offers the concluding remarks.

2. Approaches for instance selection

From the literature, it can be studied that during the past decade many approaches for instance selection or reduction have been utilized. Generally, these approaches or methods are divided into two main classes based on the criteria for selection of an instance i.e., *filters approaches* and *wrapper approaches*.

2.1. Filter approaches

Filter approaches are not concerned with classifier accuracy instead these methods are concerned with application of certain filters on the instances. Some filter methods trains the classifier on border instances of each class such as Pair Opposite Class Nearest Neighbor (POC-NN) [2] and Pattern for Ordered Projections (POP) [11]. Some of the filter methods such as clustering (CLU) [12] and Object Selection by Clustering (OSC) [13] make use of clusters centers for selection. Some uses weights of instances to select instances such as Weighing Prototypes (WP) [14] and Prototype Selection by Relevance (PSR) [15]. In [3] single layer feed forward neural network RWNN is used for reducing the size of dataset using instance selection. Shayegan et al. [4] proposed a method in which modified frequency diagram technique is used for Optical Character Recognition. A fuzzy rough instance selection approach was presented in [5]. In [8] sample entropy based dataset reduction method was proposed for ELM. A number of other methods are suggested in literature like instance selection for one class problem [16] and selection of instances in Meta Learning to estimate the performance of classifier [9]. Moreover, some authors used a combination of different instance selection techniques already present in literature for improved performances [17-19]. Several surveysare done for different instance selection methods in [9, 20-26].

2.2. Wrapper approaches

In wrapper approaches, learning algorithms are used for generation of a subset of dataset. These approaches involve accuracy of any learning algorithm to produce a subset. Dataset is reduced by discarding such instances that do not take part in improving the accuracy of learning algorithm. Most of the instance selection methods that use wrapper approaches are based on k-NN classifier.

In this paper, we used seven instance selection techniques (i.e., wrapper approaches) to analyze the effectiveness by using single layer feed-forward neural network (SLFN). The context and functionalities of IS techniques that are used in this paper are mentioned in later sections.

2.2.1. Edited nearest neighbor

Edited Nearest Neighbor (ENN) proposed by Wilson [25] decreases the size of training dataset by eliminating the noisy instances from original training dataset D. Instance that have different class

than that of its majority of neighbors is eliminated. In ENN no. of neighbors k=3, its pseudo-code is presented in Algorithm 1.

Algorithm 1 Pseudo-code of edited nearest neighbor [25]

Input: Data set $D = (x_i, c_i \mid i = 1, \ldots, N)$
Output: Reduced dataset *R*.
Process: for each *x*,

1. Find *kNN* of *x*.
2. Find *c(m)*, the class of majority of *kNN*.
3. If *c(x) != c(m)*, discard *x*.

2.2.2. All k-Nearest neighbor

All k Nearest Neighbor (All kNN) proposed by [22] is a modification of ENN. It repeats ENN for all the instances in *D* but instead of removing, it flags the noisy instances and remove all the flagged instances at once in the end, its pseudo-code is presented in Algorithm 2.

Algorithm 2 Pseudo-code of all k nearest neighbor [22]

Input: Data set $D = (x_i, c_i \mid i = 1, \ldots, N)$
Output: Reduced dataset *R*.
Process: for each *x*

1. Find *kNN* of *x*.
2. Find *c(m)*, the class of majority of *kNN*.
3. If *c(x) != c(m)*, flag *x*.

Discard all flagged instances at once.

2.2.3. Modified edited nearest neighbor

In Modified Edited Nearest Neighbor (MENN) proposed by [12] number of nearest neighbors to determine the class of an instance x is not fixed. It removes such instances form D that doesn't belong to the class of majority of its k + l nearest neighbors. Here l is the number of instances that are at the distance equal to the distance of last neighbor of *x*, its pseudo-code is presented in Algorithm 3.

Algorithm 3 Pseudo-code of modified edited nearest neighbor [12]

Input: Data set $D = (x_i, c_i \mid i = 1, \ldots, N)$
Output: Reduced dataset *R*.
Process: for each *x*

1. Find $k + l$ *NN* of *x*.
2. Find *c(m)*, the class of majority of $k + l$ *NN*
3. *If c(x)!=c(m)*, discard *x*.

2.2.4. Nearest centroid neighbor

Nearest Centroid Neighbor (NCN) [27] uses the nearest centroid neighbors for elimination of instances from the dataset. An instance that is dissimilar of majority of its nearest centroid neighbors is eliminated from D. A neighbor is nearest centroid if neighbor and mean of the neighbors both are nearest to x; its pseudo-code is presented in Algorithm 4.

Algorithm 4 Pseudo-code of nearest centroid neighbor [27]

Input: Data set $D = (x_i, c_i\ 1| \ \dot{\leq} \ N)$
Output: Reduced dataset *R*.
Process: for each *x*

1. Find *k NCN* of *x*.
2. Find *c(m),* the class of majority of *k NCN*.
3. *If* $c(x) != c(m)$, discard *x*.

2.2.5. ENN estimating class Probabilistic and Threshold

Edited Nearest Neighbor estimating class Probabilistic and Threshold (ENNTh) [24] computes weighted probabilities of *k* neighbors' class of an instance *x* and then computes the probability of *x* to belong to its class. If probability of *x* to belong to its class is lesser than a specified threshold (0 < threshold < 1), x is eliminated, its pseudo-code is presented in Algorithm 5.

Algorithm 5 Pseudo-code of ENN estimating class Probabilistic and Threshold [24]

Input: Data set $D = (x_i, c_i\ 1| \ \dot{\leq} \ N)$
Output: Reduced dataset *R*.
Process: for each *x*

1. Find *k NN of x.*
2. Compute weighted probabilities of *k NN* of *x.*
3. Calculate *P(x),* the probability of *x* to belong to its class
4. If *P(x) < (Threshold),* discard *x*.

2.2.6. Relative neighborhood graph editing

Relative Neighborhood Graph Editing (RNG) is a graph based algorithm which uses relative neighborhood graph for selection of instances [19],its pseudo-code is presented in Algorithm 6.

Algorithm 6 Pseudo-code of Relative neighborhood graph editing [19]

Input: Data set $D = (x_i, c_i\ 1| \ \dot{\leq} \ N)$
Output: Reduced dataset *R*.
Process: Draw a graph $G = (V, E)$ for all *x* in *D* where *V* =*x* in *D* and *E* is the edge between two neighbors. For each *V,*

If *V* is misclassified by its neighbors

1. Consider sub graph G_S such that G_S consists of *V* and its neighbors
2. Find *c(m),* class of majority of neighbors of G_S.
3. If *c(V) != c(m),* discard *V*.

2.2.7. MultiEdit

Multiedit proposed by Devijiver [11] randomly divides D into $S_{1..N}$ groups, it selects such instance from each group S_i and adds them to R that have different class than that of its k nearest neighbors in group $S_{(i+1)modN}$, its pseudo-code is presented in Algorithm 7.

Algorithm 7 Pseudo-code of ENN estimating class Probabilistic and Threshold [11]

Input: Data set $D = (x_i, c_i\ 1| \dot{\leq} N)$
Output: Reduced dataset R.
Process: Divide D in N partitions. For each x in each S_i,

1. Find k NN in $S_{(i+1)modN}$.
2. *If* x is misclassified by its k NN, discard x.

3. Extreme learning machine

ELM [15] is simplest form of feedforward neural network that uses only single hidden layer (SLFN) [13, 14, 18]. ELM is popular due to following main properties:

1. Its efficiency in terms of time as it is thousands of times quicker than other algorithms.
2. It ability to provide remarkable results without parameter tuning.
3. Its ability to provide great results without Back Propagation (BP) which is the main technique used by traditional neural networks for the improvement of weights.
4. Its ability to generalize the results for classification.

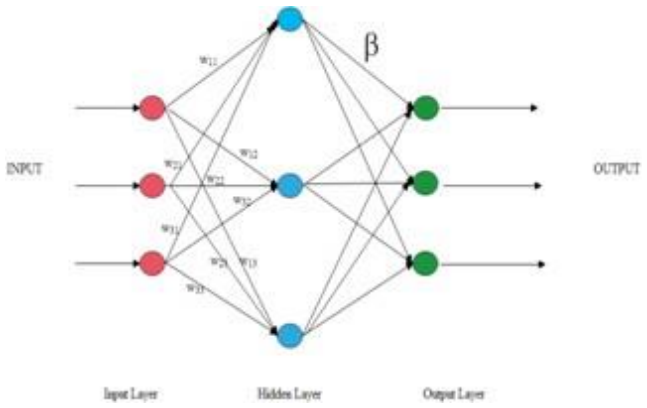

Figure 2: Extreme Learning Machine

Algorithm 8 Pseudo-code of ELM [15]

Input: Data set $D = \{(x_i, t_i)\ x_i \in R^d, t_i \in R^m\}$, Output function of hidden node $= G(a_i, b_i, x)$ and $N=$ no. of hidden nodes.
Output: Output weight vector β.
Process:

1. Generate bi and w_i, where b_i is the bias and w_i is the weight between input layer nodes and

hidden layer nodes.
2. Calculate hidden layer output matrix denoted by *H*.
3. Determine β by using the equation $\beta = H^{-1}T$ where T is the target output.

ELM has its wider applications in various fields. It is being used in image processing for facial and pattern recognition [17], lithology identification [29], time series analysis [23], image fusion after remote sensing[30] and diabetes diagnosis [15] etc. In [20] the ELM gave satisfactory results for intrusion detection system. In [21] car license plate detection system was developed using ELM. An efficient Indoor Positioning system was developed using ELM [16]. A weighted ELM was applied successfully to detect credit card fraud in [26].

4. Experimental design

14 classification datasets are selected from UCI Machine Learning Repository [1] to experimentally evaluate the performance of ELM after applying 7 different instance selection techniques (Section 2) on the datasets. Main characteristics of 14 datasets are given in Table 1. Instance reduction is done using KEEL and the performance of ELM over original and reduced data is evaluated using MATLAB. We carried out our experiment using 10 fold cross validation.

Table 1: Detail of Datasets

Dataset	Input features	Classes	Total instances
Diabetic Retinopathy Debrecen	20	2	1151
Banknote Authentication	4	2	1372
Contraceptive Method Choice	9	3	1473
Yeast	8	2	1484
Phishing Website	30	2	2456
Seismic Bumps	18	2	2584
Wine Quality White	11	7	4898
Page Blocks	10	2	5473
Electrical Grid Stability	13	2	10000
Polish Companies Bankruptcy	64	2	10503
EEG Eye State	14	2	14980
HTRU2	8	2	17898
Magic Telescope	10	2	19020
Avila	10	12	20867

Specifically, our experiment comprises of the following steps:

1) As data preprocessing is the first step to perform any data mining or machine learning task, we first prepared our datasets to get the efficient and well-organized data. Data rescaling has great impact on the performance of classification. Therefore, all training and testing datasets are rescaled within range 0 and 1 using *normalize* filter in Weka.

2) Randomly divided the datasets into training and testing sets with proportions 70% and 30%

respectively.

3) Seven different instance selection techniques are applied on training datasets using keel.
4) Reduced training datasets are fed to seven different ELM-based classifiers. Sigmoid activation function and 20 hidden neurons are used for training of each classifier.
5) For each of the reduced training dataset, computed the training times and training accuracies of above trained seven classifiers.
6) Monte Carlo method is used on each dataset and repeated steps 4 and 5 five times and computed average of training times and training accuracies.
7) A new ELM based classifier is also trained on original normalized training datasets and Monte Carlo is used for calculating training time and accuracy.
8) Evaluated the performance of all trained classifiers (trained on seven reduced training datasets and original set) on testing datasets using Monte Carlo method and noted averaged testing times and testing accuracies.

The flowchart of experiment is given in Figure 3.

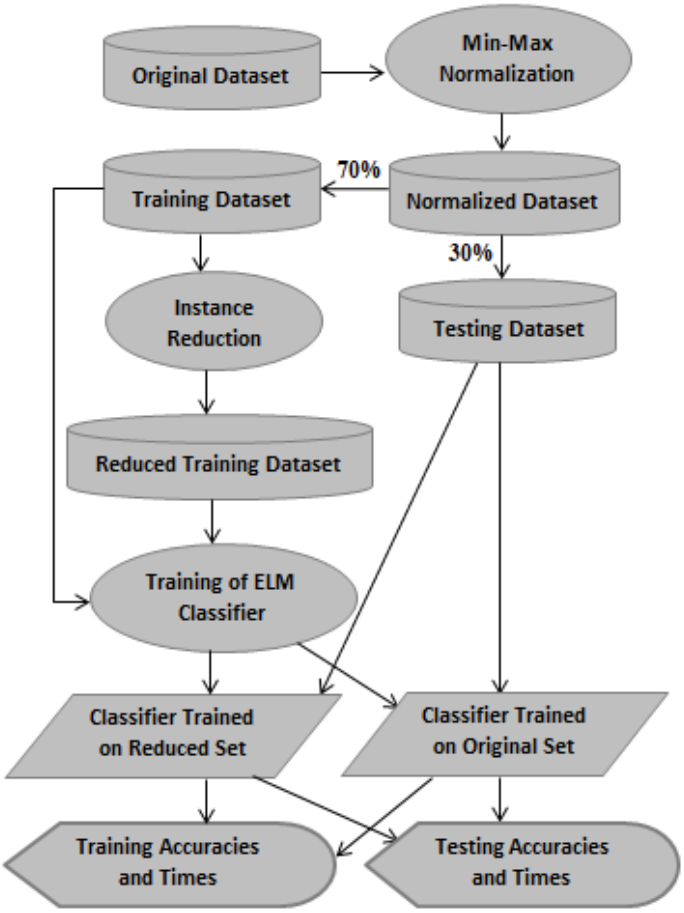

Figure 3: Flowchart of Experiment

5. Experimental Results and Analysis

Mainly our experiment consists of two phases: instance reduction of original normalized datasets and performance evaluation of ELM over reduced and original datasets.

5.1. Analysis of Instance Reduction

As discussed earlier, seven instance reduction techniques are applied over original normalized datasets. The percentage of instance reduction of each technique is examined that is summarized in Table 2 and is shown graphically in Figure 4 and Figure 5. Results depict that MENN and ENNTh has more reduction percentage in contrast to other instance selection techniques.

- MENN has the highest percentage of reduction in most datasets which proves that when algorithm (MENN) is allowed to vary the value of k depending on the dataset to be reduced, it gives remarkable results.
- ENNTh has second highest percentage of reduction which indicates that use of threshold on weighted probabilities of neighbors of an instance for deciding its fate also reduces the datasets to a great extent.
- AllKNN and Multiedit are also good at reducing the size of dataset which depicts that nearest neighbors are also good identifiers of an instance class.

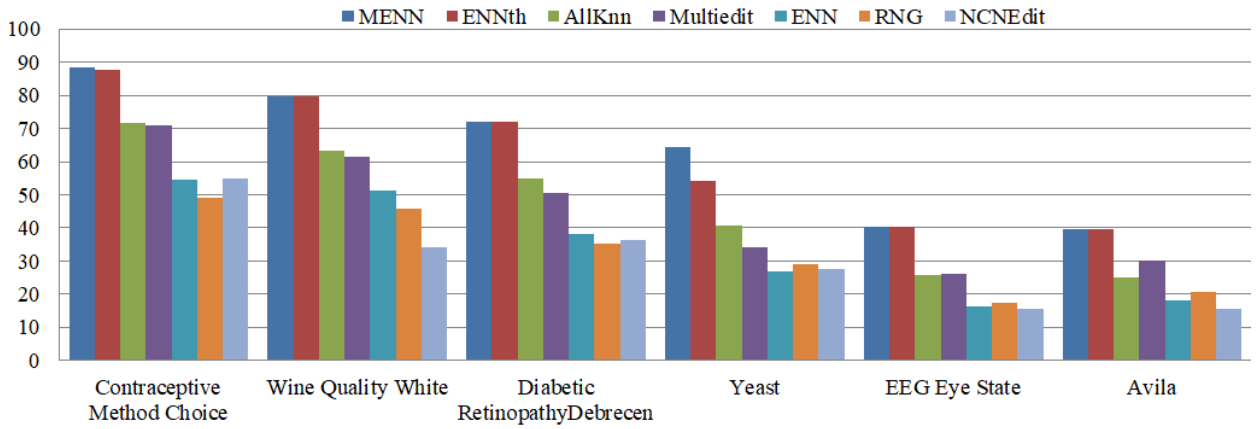

Figure 4: Instance reduction in 6 datasets against seven techniques

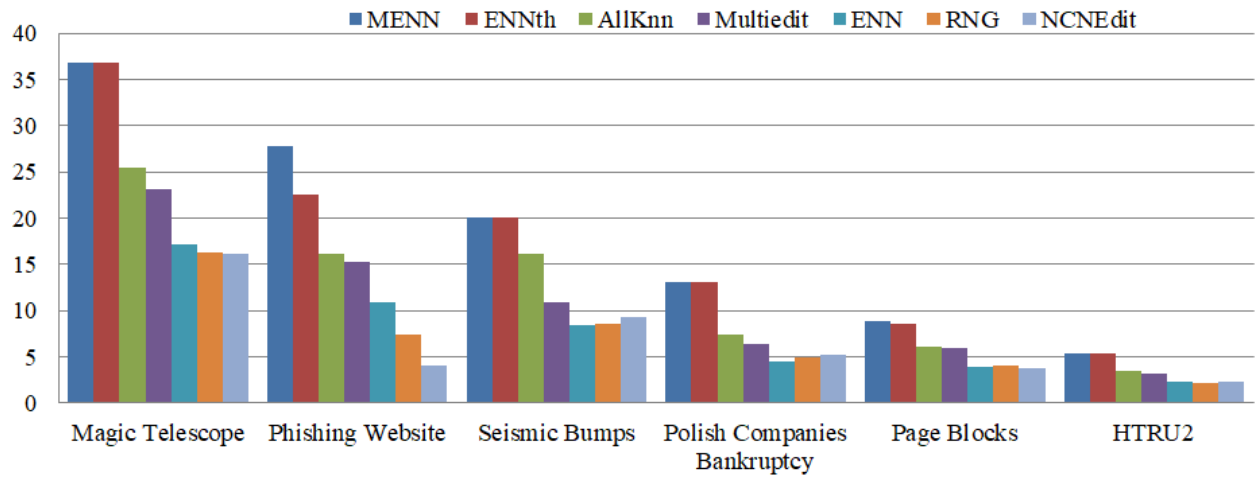

Figure 5: Instance reduction in 6 datasets against seven techniques

Table 2: Percentage of Instance Reduction in Datasets against Each Reduction Technique

Dataset	**AllKnn**	**RNG**	**ENN**	**MENN**	**Multiedit**	**ENNth**	**NCNEdit**
Avila	25.11	20.50	18.14	39.70	30.30	39.60	15.70
Banknote Authentication	0.20	0.60	0.20	0.20	0.30	0.20	0.20
Contraceptive Method Choice	71.80	49.20	54.70	88.30	71.00	87.60	55.00
Diabetic RetinopathyDebrecen	54.80	35.10	38.00	72.20	50.40	71.90	36.40
Electrical Grid Stability	19.20	8.20	9.80	30.30	16.30	30.30	9.00
EEG Eye State	25.70	17.20	16.40	40.39	26.20	40.39	15.70
HTRU2	3.40	2.16	2.24	5.40	3.09	5.40	2.24
Magic Telescope	25.50	16.30	17.10	36.80	23.10	36.80	16.20
Page Blocks	6.00	4.10	3.90	8.90	5.90	8.60	3.80
Phishing Website	16.20	7.40	10.90	27.80	15.20	22.60	4.10
Polish Companies Bankruptcy	7.39	4.88	4.47	13.10	6.30	13.10	5.23
Wine Quality White	63.30	45.70	51.30	79.60	61.30	79.60	34.20
Seismic Bumps	16.20	8.60	8.40	20.10	10.90	20.10	9.30
Yeast	40.70	29.00	26.80	64.50	34.00	54.00	27.70

5.2. Analysis of ELM's Performance

In the second phase of experiment, we further validated the effectiveness of instance reduction on the performance of ELM in terms of time and accuracy. We used Monte Carlo method and computed averaged times and accuracies of each original and reduced dataset against each reduction method. It was noticed that training and testing time of ELM over original dataset was already insignificant i.e. less than a second for even larger datasets like Avila (10 attributes & 20867 instances), Magic telescope (10 attributes & 19020 instances), EEG Eye State (10 attributes & 14980 instances) and Polish companies bankruptcy (64 attributes & 10503 instances). After dataset reduction using seven instance selection techniques training and testing time further decreases.

Training and testing accuracies of ELM classifiers trained on original and reduced datasets corresponding to seven instance reduction techniques is summarized in Table 3 and Table 4. It is observed that:

- Training accuracy of every original dataset is lower than training accuracy of each reduced training dataset.
- Datasets reduced using MENN and ENNTh resulted in best training accuracy.
- Similarly, testing accuracy every original dataset is lower than accuracy of same dataset when it is tested on classifiers trained on reduced datasets.
- Datasets decreased by ENNTh produced greatest testing accuracy.
- Testing accuracy of datasets reduced by MENN is also greater in some datasets.

This verifies that performance of ELM is boosted in terms of training and testing accuracy after instance reduction as shown graphically in Figures 6, 7, 8 and 9.

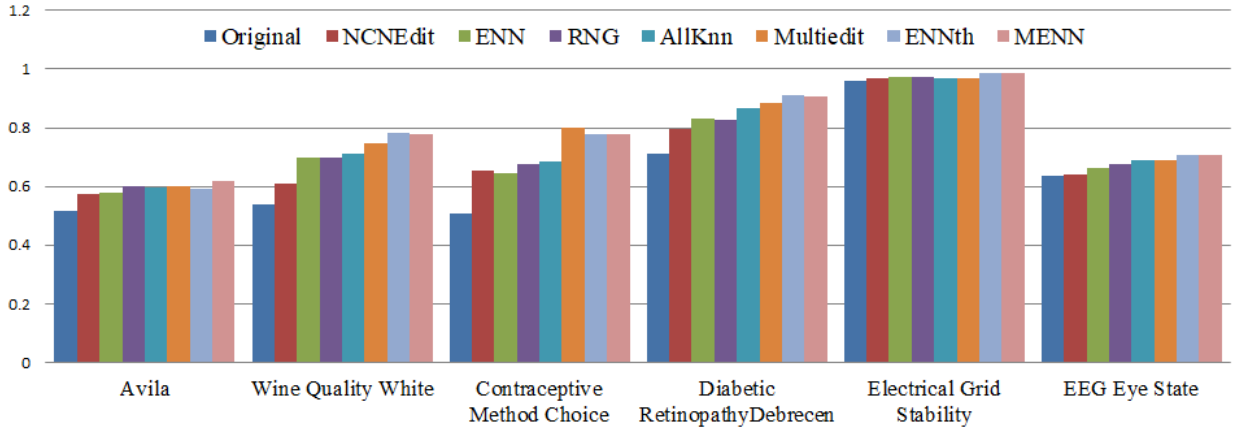

Figure 6: Training accuracy of original and seven reduced datasets

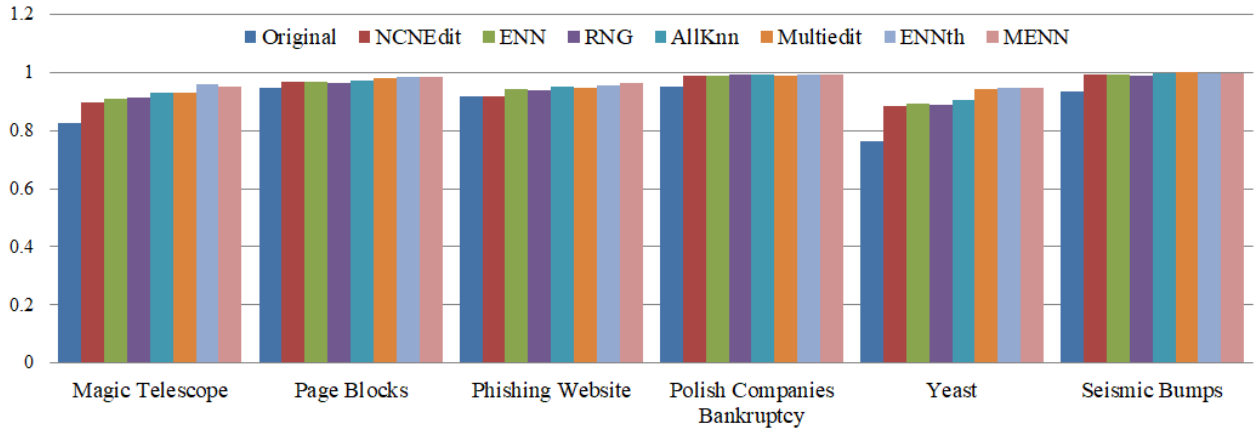

Figure 7: Training accuracy of original and seven reduced datasets

Table 3: Training Accuracy of Original and Reduced Datasets

Dataset	Original	NCNEdit	ENN	RNG	AllKnn	Multiedit	ENNth	MENN
Avila	0.5174	0.5737	0.5767	0.6000	0.5966	0.6011	0.5933	0.6194
Banknote Authentication	0.9953	0.9903	0.9953	0.9962	0.9943	0.9893	0.9943	0.9942
Contraceptive Method Choice	0.5093	0.6546	0.6451	0.678	0.6846	0.7995	0.7795	0.7795
Diabetic Retinopathy Debrecen	0.71	0.7972	0.8306	0.8256	0.8688	0.8842	0.909	0.9084
Electrical Grid Stability	0.9587	0.9677	0.9748	0.9718	0.9673	0.9681	0.9879	0.9862
EEG EyeState	0.6355	0.6423	0.6614	0.675	0.6873	0.6904	0.7054	0.7059
HTRU2	0.9739	0.9882	0.9879	0.9887	0.9898	0.9915	0.9921	0.9926
Magic Telescope	0.825	0.8983	0.9093	0.9136	0.9301	0.9324	0.9578	0.9511
Page Blocks	0.9455	0.9682	0.9694	0.9645	0.9724	0.9803	0.9859	0.9847
Phishing Website	0.9185	0.9164	0.9446	0.9406	0.9525	0.9478	0.9564	0.965
Polish Companies Bankruptcy	0.9528	0.9872	0.9894	0.992	0.9919	0.9911	0.9928	0.9928
Wine Quality White	0.5399	0.6108	0.696	0.6977	0.7107	0.7463	0.7802	0.7764
Seismic Bumps	0.9339	0.9915	0.9948	0.9892	0.9958	1.0000	0.9989	0.9989
Yeast	0.7649	0.8839	0.8916	0.8888	0.9068	0.9427	0.9491	0.9471

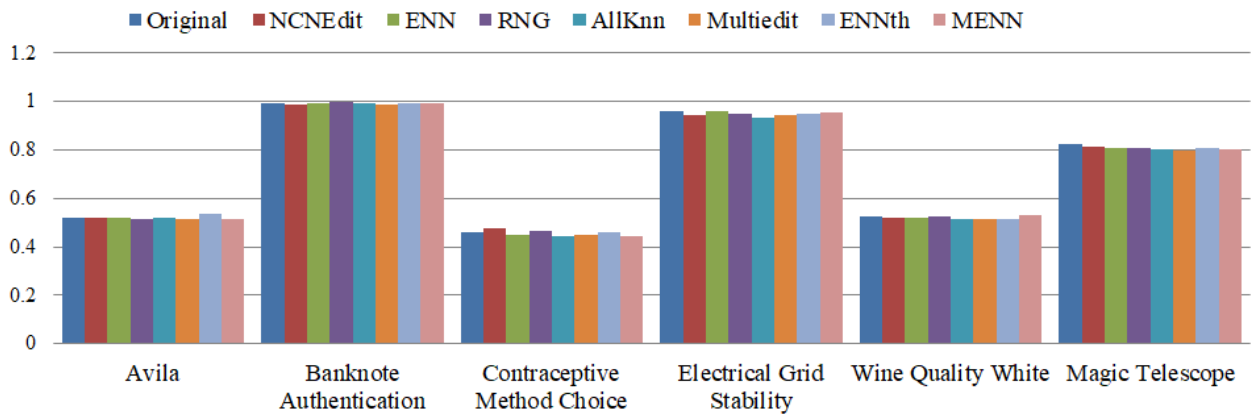

Figure 8: Testing accuracy of original and seven reduced datasets

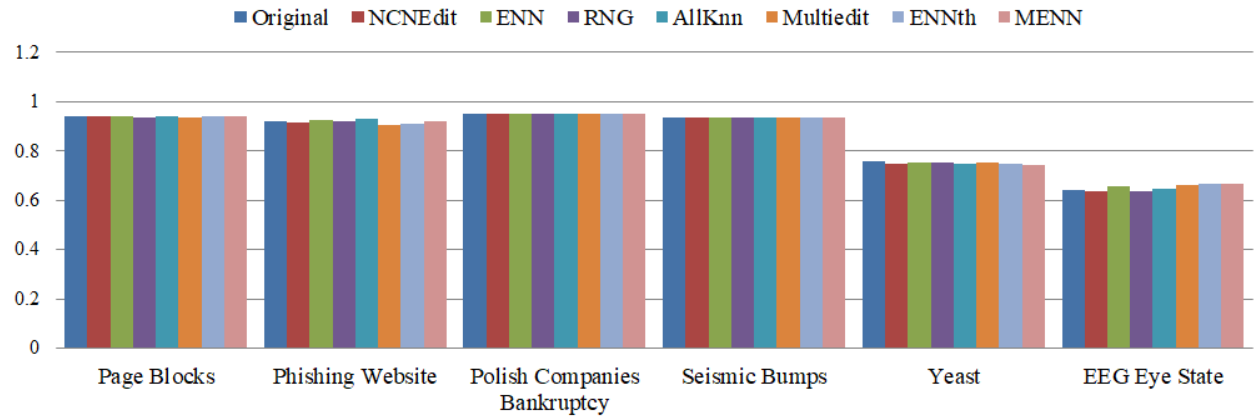

Figure 9: Testing accuracy of original and seven reduced datasets

Table 4: Testing Accuracy of Original and Reduced Datasets

Dataset	Original	NCNEdit	ENN	RNG	AllKnn	Multiedit	ENNth	MENN
Avila	0.5194	0.5208	0.5207	0.515	0.5223	0.5127	0.5337	0.5169
Banknote Authentication	0.9942	0.9884	0.9942	0.9971	0.9928	0.987	0.9943	0.9942
Contraceptive Method Choice	0.4608	0.477	0.4513	0.4676	0.4459	0.4513	0.4622	0.4446

Diabetic Retinopathy Debrecen	0.7224	0.6845	0.6983	0.6776	0.6655	0.6707	0.6552	0.6655
Electrical Grid Stability	0.9576	0.9412	0.96	0.9514	0.9328	0.9412	0.9488	0.9536
EEG EyeState	0.641	0.6356	0.6593	0.6376	0.6479	0.6602	0.6659	0.667
HTRU2	0.9733	0.9722	0.9723	0.9726	0.9722	0.972	0.9716	0.9719
Magic Telescope	0.8237	0.8138	0.8083	0.8066	0.8016	0.7994	0.8091	0.8049
Page Blocks	0.9398	0.9394	0.9387	0.9354	0.9423	0.9354	0.9401	0.9391
Phishing Website	0.9228	0.9163	0.9276	0.9203	0.9293	0.9065	0.9106	0.9203
Polish Companies Bankruptcy	0.9517	0.952	0.9522	0.9522	0.9524	0.9522	0.952	0.9524
Wine Quality White	0.5241	0.5192	0.5171	0.5261	0.5143	0.5139	0.5167	0.5318
Seismic Bumps	0.9344	0.9344	0.9344	0.9344	0.9344	0.9344	0.9344	0.9344
Yeast	0.7571	0.7477	0.7517	0.7544	0.7477	0.753	0.749	0.7423

6. Conclusion

In this experimental study, we verified the classification performance of extreme learning after applying different instance reduction techniques.14 different datasets having different number of classes and different dimensions were downloaded from the UCI machine learning repository. We examined that the reduction percentage of ENNTh and MENN is greater than other techniques. The performance of ELM classifier trained separately on original training dataset and on reduced training datasets was also examined in terms of time and accuracy. We noted down the training time, training accuracy, testing time and testing accuracy of trained models after testing them on testing dataset. Final results show that ELM is already very efficient in terms of training and testing times so instance selection has almost zero influence on training and testing times but improvement in training accuracies and testing accuracies is evident after instance selection. We concluded that efficiency of ELM enhances after elimination of noisy instances from the datasets. This study can provide useful insights and instructions for research community and practical applications of ELM.

References

[1] Doina Caragea, Adrian Silvescu, and Vasant Honavar. *A framework for learning from distributed data using sufficient statistics and its application to learning decision trees*. International Journal of Hybrid Intelligent Systems, 1(1-2):80–89, 2004.

[2] William J Frawley, Gregory Piatetsky-Shapiro, and Christopher J Matheus. *Knowl- edge discovery in databases: An overview*. AI magazine, 13(3):57, 1992.

[3] Huan Liu and Hiroshi Motoda. *Instance selection and construction for data mining*, volume

608. Springer Science & Business Media, 2013.
[4] Huan Liu, Hongjun Lu, and Jun Yao. *Identifying relevant databases for multidatabase mining*. pages 210–221. 1998.
[5] Huan Liu and Hiroshi Motoda, editors. *Instance Selection and Construction for Data Mining*. Springer US, Boston, MA, 2001.
[6] Ireneusz Czarnowski and Piotr Jedrzejowicz. *An Approach to Instance Reduction in Supervised Learning*. In Research and Development in Intelligent Systems XX, pages 267–280. Springer London, London, 2004.
[7] Jose´ Ramo´n Cano, Francisco Herrera, and Manuel Lozano. *On the combination of evolutionary algorithms and stratified strategies for training set selection in data mining*. Applied Soft Computing, 6(3):323–332, mar 2006.
[8] V. Karunakaran, M. Suganthi, and V. Rajasekar. *Feature selection and instance se- lection using cuttlefish optimisation algorithm through tabu search*. International Journal of Enterprise Network Management, 11(1):32, 2020.
[9] J Arturo Olvera-Lo´pez, J Ariel Carrasco-Ochoa, J Francisco Mart´ınez-Trinidad, and Josef Kittler. *A review of instance selection methods*. Artificial Intelligence Review, 34(2):133–143, 2010.
[10] Ireneusz Czarnowski and Piotr Jedrzejowicz. *An Approach to Instance Reduction in Supervised Learning*. In Research and Development in Intelligent Systems XX, pages 267–280. Springer London, London, 2004.
[11] Pierre A Devijver. *On the edited nearest neighbor rule*. In Proc. 5th Int. Conf. Pattern Recognition, 1980, 1980.
[12] Kazuo Hattori and Masahito Takahashi. *A new edited k-nearest neighbor rule in the pattern classification problem*. Pattern Recognition, 33(3):521–528, 2000.
[13] Guang-Bin Huang and Haroon A Babri. *Upper bounds on the number of hidden neu- rons in feedforward networks with arbitrary bounded nonlinear activation functions*. IEEE transactions on neural networks, 9(1):224–229, 1998.
[14] Guang-Bin Huang, Yan-Qiu Chen, and Haroon A Babri. *Classification ability of single hidden layer feedforward neural networks*. IEEE transactions on neural net- works, 11(3):799–801, 2000.
[15] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. *Extreme learning machine: theory and applications*. Neurocomputing, 70(1-3):489–501, 2006.
[16] Xiaoxuan Lu, Han Zou, Hongming Zhou, Lihua Xie, and Guang-Bin Huang. *Robust extreme learning machine with its application to indoor positioning*. IEEE transac- tions on cybernetics, 46(1):194–205, 2015.
[17] Yong Peng, Suhang Wang, Xianzhong Long, and Bao-Liang Lu. *Discriminative graph regularized extreme learning machine and its application to face recognition*. Neurocomputing, 149:340–353, 2015.
[18] Tariq Rashid. *Make your own neural network: a gentle journey through the math- ematics of neural networks, and making your own using the Python computer language*. CreateSpace Independent Publ., 2016.
[19] Jose´ Salvador Sa´nchez, Filiberto Pla, and Francesc J Ferri. *Prototype selection for the*

nearest neighbour rule through proximity graphs. Pattern recognition letters, 18(6):507–513, 1997.

[20] Gideon Creech and Frank Jiang. *The application of extreme learning machines to the network intrusion detection problem*. In AIP Conference Proceedings, volume 1479, pages 1506–1511. American Institute of Physics, 2012.

[21] Sumanta Subhadhira, Usarat Juithonglang, Paweena Sakulkoo, and Punyaphol Ho- rata. *License plate recognition application using extreme learning machines*. In 2014 Third ICT International Student Project Conference (ICT-ISPC), pages 103–106. IEEE, 2014.

[22] Ivan Tomek. *An experiment with the edited nearest-neighbor rule*. IEEE Transac- tions on Systems, Man, and Cybernetics, 6(6):448–452, 1976.

[23] Mark Van Heeswijk, Yoan Miche, Tiina Lindh-Knuutila, Peter A J Hilbers, Timo Honkela, Erkki Oja, and Amaury Lendasse. *Adaptive ensemble models of extreme learning machines for time series prediction*. In International Conference on Artifi- cial Neural Networks, pages 305–314. Springer, 2009.

[24] Fernando Va´zquez, J Salvador Sa´nchez, and Filiberto Pla. *A stochastic approach to Wilson's editing algorithm*. In Iberian Conference on Pattern Recognition and Image Analysis, pages 35–42. Springer, 2005.

[25] Dennis L Wilson. *Asymptotic properties of nearest neighbor rules using edited data*. IEEE Transactions on Systems, Man, and Cybernetics, (3):408–421, 1972.

[26] Honghao Zhu, Guanjun Liu, Mengchu Zhou, Yu Xie, Abdullah Abusorrah, and Qi Kang. *Optimizing Weighted Extreme Learning Machines for Imbalanced Classi- fication and Application to Credit Card Fraud Detection*. Neurocomputing, 2020.

[27] B B Chaudhuri. *A new definition of neighborhood of a point in multi-dimensional space*. Pattern recognition letters, 17(1):11–17, 1996.

[28] https://archive.ics.uci.edu/ml/datasets.php

[29] Cai, Lei, Guo-Jian Cheng, and Hua-Xian Pan. *Lithologic identification based on ELM*. Computer Engineering and Design 31.9 (2010): 2010-2012.

[30] Yao, Wei, and Min Han. *Fusion of thermal infrared and multispectral remote sensing images via neural network regression*. J Image Graphics 15.8 (2010): 1278-1284.