VOL. 14, NO. 1, JANUARY 2019 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

A NEW FORMULA TO DETERMINE THE OPTIMAL DATASET SIZE FOR TRAINING NEURAL NETWORKS

Lim Eng Aik[1], Tan Wei Hong[2] and Ahmad Kadri Junoh[1]
[1]Institut Matematik Kejuruteraan, Universiti Malaysia Perlis, Arau, Perlis, Malaysia
[2]School of Mechatronic Engineering, Universiti Malaysia Perlis, Arau, Perlis, Malaysia
E-Mail: e.a.lim80@gmail.com

ABSTRACT

In neural networks, training a network with a large datasets put a heavy load to computation time and does not guarantee networks accuracy. As dataset may contains outlier or missing value that leave a gap that possibly cause the overall shape of dataset to be affected during training session. A datasets with too limited data points or too much data points is not an optimal size for training the neural network. Hence, suitable size is requires ensuring the neural network is trained using optimal dataset size which able to reduce computational time and does not affect the accuracy significantly. This paper presents a dataset size reduction formula that can provide suitable number of training dataset size for the neural networks and does not affect the accuracy significantly. The formula derived from the Fibonacci retracement that has been reported its usage in many literatures. The experiments were performed on four literatures function and four real-world datasets to validate its efficiency. The experiments tested on groups of dataset with their data reduce from 0 percent to 95 percent with 5 percent step size. The results are compared to proposed method for root mean square error (RMSE) and time usage in radial basis function network (RBFN). The proposed method yielded a promising result with an average over 50 percent reduction in time usage and 20 percent in RMSE.

Keywords: neural networks, dataset size reduction, training data, radial basis function network.

1. INTRODUCTION

Neural networks (NN) as an approximators need to go through a series of learning process, consists of different steps such as; building a training set, training the network system, then testing its curve behavior, and finally approximate the unknown values. During training process, NN using a distance based rule as approximator, i.e. Euclidean distance, where each sample points are calculated in relative to a set of random center select from the sample dataset. The main problem is that the distance calculation requires long time to connect each sample points to each center during training process. Such problem occur due to NN rule presents some space deficiency when the number of input vectors reaches the available resources, since all samples are stored in a memory. Hence, to fix such deficiency, such large number of training set can be reduced, and this would shorten the time taken for each new distance calculation, while minimize the resources memory usage.

This paper focused on reducing the training set size without inflicting a significant loss in approximation accuracy. The formula proposed in this paper aims to reduces training dataset size while maintaining sufficient samples that significantly interpret the form or dispersion of a model of the problem. A new formula that able to reduce the sizing of a training dataset for NN while maintaining similar levels of accuracy in approximation results equally to entire training set is applied. The proposed formula reduce the number of training dataset size by calculating the total training dataset size with Fibonacci retracement ratio and sum it with a bias size. The experiments are performed on 4 functions from literatures which consist of 1-dimension to 5-dimensions inputs, and 4 real-world datasets. The experimental results present that the formula able to significantly reduce the sizing of a training dataset while maintaining similar or better accuracy of approximation as when the whole training dataset is used.

2. RELATED WORKS

One of the best features of neural networks is its ability to generalize and approximate a sample data without the need of specify equation and coefficients, particularly when an unknown model describing an unknown complex relation and training data abundant. Due to their ability to generalize substantially, Radial Basis Function networks (RBFN) are usually selected for this purpose (Arteaga & Marrero, 2013; Kavaklioglu, Koseoglu, & Caliskan, 2018; Lin, 2016; Majdisova & Skala, 2017; Smolik, Skala, & Majdisova, 2018). Furthermore, in this big data era, many domains such as image processing, text categorization, biometric, microarray, etc. had the size of datasets so large, that real-time system requires long time and memory storage to process them.

Under such conditions, approximation task using available datasets can become a challenging task and difficult. This problem is more challenging in distance based learning algorithms such as neural network (Albalate, 2007; Haykin, 1994; Smolik *et al.,* 2018), nearest neighbor (B. Dasarathy, 2002; B. V. Dasarathy, 1991), clustering method (Jędrzejowicz & Jędrzejowicz, 2016; Kirsten & Wrobel, 1998; Liu, Zhang, Zhang, & Cui, 2017) and support vector machine (Dahiya, 2014; Sebtosheikh & Salehi, 2015; Wang, Li, Liu, Zhang, & Zhang, 2014).

By default, the NN algorithm must search through all available training samples which requires large memory, and performs distance to center calculation, which is slow during training of NN for approximation

purposes. Additionally, due to NN stores all samples distances for training datasets, thus, noise distances are stored as well, which can cause degrade in approximation accuracy. In early 90s, a team of researcher led by Foody (FOODY, McCULLOCH, & YATES, 1995) study the effects of training set size on NN classification accuracy. Their finding shows higher classification accuracy does not need large dataset, instead, NN only need important data samples that can represent the overall shape or picture of the case. This fact is supported by Zhou et al. (Zhou, Wei, Li, & Dai, 2004) and Roy *et al.* (Roy, Leonard, & Roy, 2008) in their literatures that mentioned the accuracy of model during training does not increase along the increase of training set size (Ougiaroglou, Diamantaras, & Evangelidis, 2017).

From enormous proposals to tackle this problem, conventional methods rely on removing some training datasets, in which they refer as dataset reduction. From the 90s, Gerardo and Perez (Gerardo & Perez, 1998) proposed a stratified sampling approach to reduce training dataset size. However, their approach does not have mathematical algorithm for further improvement. In early 20th century, Lozano team (Lozano, Sánchez, & Pla, 2003) reported the approach of dataset reduction using NCN-based exploratory procedures. The approach obtained strongly reduced training dataset with good accuracy in classification test. Sanchez (Sánchez, 2004) in his independent research, discovered a new approach to reduce training dataset using prototype generation and space partitioning that yield good results. However, the algorithm involves complicated calculation that is not easily understandable by regular researcher.

In a literature by (Ougiaroglou & Evangelidis, 2012), proposed an effective data reduction algorithm that could lower the preprocessing cost and memory requirement with no significant change in accuracy using homogeneous clusters method. The algorithm of this method go through several process such as classify the centroid, calculating mean vector to each centroid, and apply k-means clustering to obtained the requires dataset size. In the same period, Juan *et al*. (Rico-Juan & Iñesta, 2012) also proposed method that uses nearest neighbor ranking method for selecting best training dataset samples. This approach requires multiple run in several algorithms to obtain the reduced datasets. Wang et al. (Wang et al., 2014) proposed a bootstrap sampling based data cleaning to reduce dataset size. The approach manage to reduce support vector machine training time, however, the process time requires to reduce the dataset size is not consider in the overall operation. In literature by Mohsen *et al*. (Mohsen, Kurban, Jenne, & Dalkilic, 2014), proposed random forest approach to reduce the training dataset size. The algorithm involves display similar behavior as clustering algorithm but with more rules in compared to common clustering algorithm such as k-means clustering algorithm. Dahiya (Dahiya, 2014) and Shayegan *et al.* (Shayegan & Aghabozorgi, 2014) proposed using support vector in reducing training datasets size.

In conventional use learning and simulation cases, many practitioners believe increasing the training datasets size amends the performance of learning algorithm and accuracy. It is proven that the phenomenon is not true in general for any learning algorithm and data distribution (Yousef & Kundu, 2014). This finding motivated our work and keeps us right on track to obtain the best size of dataset, a formula is necessities.

Fuangkhon team reported an algorithm for selecting the best sample points for representing the shape of a curve for approximation problem applying least boundary vector distance selection (Fuangkhon & Tanprasert, 2014). This algorithm yield good approximation accuracy, but the procedures of calculation is complicated to accomplish. Moreover, Chatchai proposed a more simple way to reduce training dataset by using Geometric Median calculation (Kasemtaweechok, 2015). However, this approach not suitable for NN training as it reduces about 90% of the datasets size. Too little training dataset is not suitable for approximation use (Ougiaroglou *et al.*, 2017; Yousef & Kundu, 2014). Lastly, Varin *et al.* (Chouvatut, Jindaluang, & Boonchieng, 2015) proposed optimum path forest method for dataset reduction. However, this approach is only applicable for classification case and not for approximating a data model.

All the literatures mentioned focus only on algorithms for reducing the dataset size and selecting the best point from it. None of the literatures proposed a mathematical formula that can directly determine the optimal datasets size without have to gone through complicated algorithm. Hence, in this paper, the proposed formula applies to approximation or prediction problems, that is, problems that require the dependent variable be predicted based on the values of independent variable.

The next section begins describing the formulation forming of proposed formula. In section 4, the proposed formula is applied in literatures function and real-world datasets, then the reduce datasets are used in RBFN for approximation accuracy test. The results of RBFN are presented and discussed. Section 5 conclude the findings and discussed some future work that would help in improving the mathematical formula.

3. THE PROPOSED METHODS

3.1 Data reduction formula

Fibonacci numbers often appear in many aspect of nature e.g. organization of leaves on a stem, shell formation, the branching of trees and our finger length. Fibonacci sequences and ratio also applied in many scientific fields such as stock trading (Bhattacharya & Kumar, 2006; Brown, 2012; Gaucan, 2011; MacLean, 2005), machine learning (Iqbal, Ghazali, & Shah, 2018) and statistical analysis (Kumar, 2014; Naka, Ino, & Kohmoto, 2005).

In term of mathematic, Fibonacci ratio is obtained by dividing any Fibonacci number by the Fibonacci number one place higher in the sequence, that is,

$$r_n = \frac{F_{n+1}}{F_n} \quad (1)$$

with the general term of Fibonacci sequence as derived by Gaucan *et al.* (Gaucan, 2011) given as,

$$F_n = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n, \quad n \geq 0 \quad (2)$$

and when $n \to \infty$ from equation (1), then the value of ratio obtain is,

$$\phi = \lim_{n\to\infty} r_n = \lim_{n\to\infty} \frac{F_{n+1}}{F_n} = \frac{1+\sqrt{5}}{2} = 1.618$$

Thus, the common Fibonacci ration is derived as

$$F_g = \frac{1}{\phi} = 0.618$$

F_g is the Fibonacci key ratio which also refer as golden ratio, obtain by dividing any number in the sequence by the number that immediately follows it. Hence, from the literatures by Gaucan *et al.* (Gaucan, 2011), all the other ratio identified are 0.236, 0.382, 0.500, 0.618 and 0.784.

Based on the ratio and from literature findings by Iqbal et al. and Bhattacharya and Kumar (Bhattacharya & Kumar, 2006; Iqbal *et al.*, 2018), that the data retrace accordance to Fibonacci ratio, we proposed a formula for data reduction S_{data} as follows:

$$S_{data} = F_{ratio} N_{data} + \Phi_{data} \quad (3)$$

where F_{ratio} represent the Fibonacci ratio mentioned, N_{data} as the total number of dataset, and Φ_{data} is the number dataset for marginal correction during Fibonacci retracement in the dataset.

Generally, the training dataset for neural networks consists of predictor and responder. Predictor is used to predict the responder value. According to findings in literatures (Brenner & Maier-Paape, 2016; Chih-Wei Hsu Chih-Chung Chang & Lin, 2010; Loh Wei-Yin & Yu-Shan Shih, 1997), the best splits used for dataset marginal correction is 5. Hence, Φ_{data} is formulate as,

$$\Phi_{data} = \frac{N_{data}}{5A_{data}} \quad (4)$$

where A_{data} is the number of predictor attributes in datasets. The responder attribute is not consider in A_{data}.

In summary, the final form of the formula for data reduction S_{data} after substitute equation (4) into equation (3) can be expressed as follows:

$$S_{data} = F_{ratio} N_{data} + \frac{N_{data}}{5A_{data}} \quad (5)$$

3.2 Radial basis function network

The radial basis function network (RBFN) basically have three layers (Broomhead & Lowe, 1988; Lei, Ding, & Zhang, 2015). The first layers utilize the input patterns and connects the networks to its environment. The hidden layer, which is the second layer in the network, and its neurons hold radial basis activation functions. Lastly, the output layer of a RBFN is equal to the weighted sum of the hidden neurons responses, which is expressed as follows:

$$y_j = \sum_{i=1}^{n} w_{ij} \varphi_i \left(\| x - c_i \| \right) + b_{0j}, \quad j = 1, 2, 3, ..., n \quad (6)$$

where n is the number of hidden nodes, x is the input vector; c_i represent the center of the i-th hidden node; w_{ij} is the weight of the i-th hidden node; φ_i is the radial basis function with c_i as its center; and b_{0j} represent the j-th node of output layer.

Note that from the input to hidden layer, the mapping is nonlinear, whereas it is linear from hidden to output layer. Moreover, Gaussian function (equation (7)) was use as the radial basis function with a simple Euclidean distance was utilized for determining the input-to-center distance.

$$\varphi_i(x) = e^{-\frac{\|x-c_i\|^2}{\sigma^2}} \quad (7)$$

where x represent input vector; c_i is the center of the i-th hidden node, and $\sigma > 0$ is the spread value, by default is predefined as 1.

Here, the RBFN was trained via NEWRB function found in MATLAB based on Demuth (Demuth, 2002) syntax as given below:

net = newrb(P,T,GOAL,SPREAD,MN)

where P,T,GOAL,SPREAD, MN and DF each represents the input or predictor vector, output or response vector, error goal, spread value of $\varphi_i(x)$ and maximum number of neurons and neuron numbers, respectively. All training for RBFN was in default values setting in predefined in NEWRB.

The performance of the RBFN for different types of dataset and dataset sizes was assessed and discussed in

following section. The performance of the created RBFN model with the new unseen data was assessed with two calculated values. Root mean square error (RMSE) was used to evaluate the prediction accuracy and express the average model prediction error (equation (8)), and the time taken for training the RBFN to show the speed of computation involves with dataset sizes.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(O_i - P_i\right)^2} \tag{8}$$

where n is the number of predicted responder; O_i is the target value for time-step i, and P_i is the predicted value of the model at time-step i.

4. RESULTS AND DISCUSSIONS

The proposed formula was tested using 4 nonlinear function from literatures, which are Santner *et al.* (Santner, Williams, & Notz, 2003) function given in equation (9), Lim *et al.* (Lim, Sacks, Studden, & Welch, 2002) function in equation (10), Dette and Pepelyshev (Dette & Pepelyshev, 2010) function in equation (11), and Friedman (Friedman, Adaptive, & Splines, 1991) function in equation (12). For all these 4 functions, the training set for RBFN consists of 400 sets of random generated data points and test set comprises of 400 sets of random generated data points, both in range of [0,1].

$$f(x) = \exp(-1.4x)\cos(3.5\pi x),\ x \in [0,1] \tag{9}$$

$$f(x) = \frac{1}{6}\left[\left(30 + 5x_1\sin\left(5x_1\right)\right)\left(4 + \exp\left(-5x_2\right)\right) - 100\right],\ x_i \in [0,1],\ \forall i = 1,2. \tag{10}$$

$$f(x) = 4\left(x_1 - 2 + 8x_2 - 8x_2^2\right)^2 + \left(3 - 4x_2\right)^2 + 16\sqrt{x_3 + 1}\left(2x_3 - 1\right)^2,\ x_i \in [0,1],\ \forall i = 1,2,3. \tag{11}$$

$$f(x) = 10\sin\left(\pi x_1 x_2\right) + 20\left(x_3 - 0.5\right)^2 + 10x_4 + 5x_5,\ x_i \in [0,1],\ \forall i = 1,2,3,4,5. \tag{12}$$

The proposed formula also was tested on 4 real-world datasets for its performance. The Biochemical Oxygen Demand (BOD) concentration dataset, phytoplankton growth rate and death rate dataset and Texas air pollutant dataset were obtained from Aik and Zainuddin (Aik & Zainuddin, 2008). Another which is the dataset from forex for EURUSD pairs is collected from XM Metatrader 4 database (XM Global, 2018).

The BOD dataset and phytoplankton dataset both consists of 100 sets of data and the test set comprises of 100 sets of data. Meanwhile, for Texas air pollutant dataset, the training set has 480 sets of data and the test set has 72 sets of data which both were taken from hourly air data. For EURUSD pairs, the training set consists of 519 sets of data taken from year 2016 to end of year 2017. The test set consists of 155 sets of data taken from early year 2018 to August 2018.

The experiments was designed by reducing the dataset size from the total training dataset size according to percentage start from 0 to 95 percent reduction of dataset, with the percentage step size of 5 percent each. Then, the proposed formula is tested with different Fibonacci ratio as mentioned in Section 3.1. Each Fibonacci ratio is label "F_ratio" format so one can easily identified the ratio used for the test.

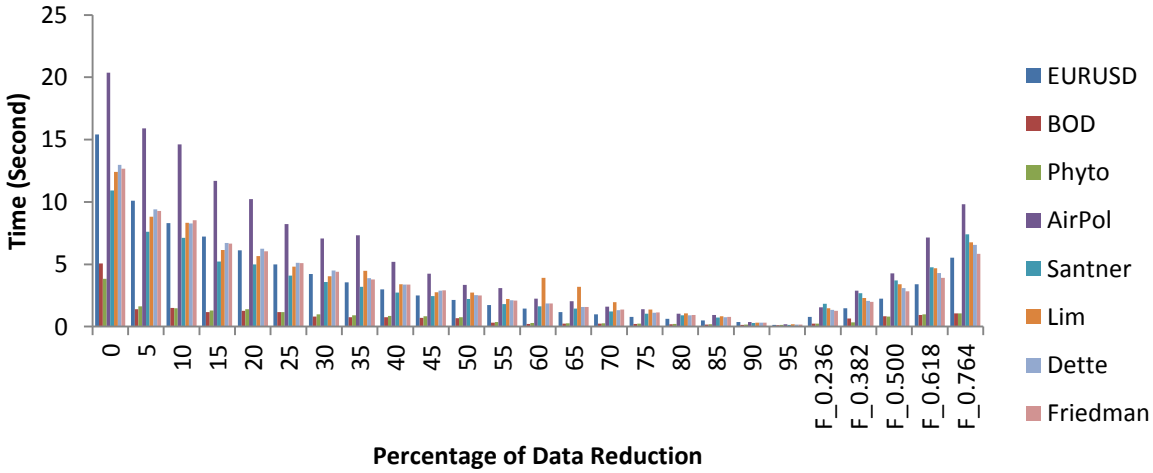

Figure-1. Overall time (in second) usage for RBFN training for each dataset.

In Figure-1, it is observed that the time usage for training RBFN decreases with bigger data reduction. Meanwhile, the increase in Fibonacci ratio obtained higher time usage was observed. Clearly, the reduction in time is significant with data size shrinkage. However, the difference is minor after percentage of data reduction reach over 50 percent. All five Fibonacci ratios with the proposed method outperformed the 20 to 0 percent reduction groups. Moreover, even the most poorly performed in the proposed method group that is, F_0.764, obtained significant reduction in time about 51.77 percent compared to full data used in training RBFN.

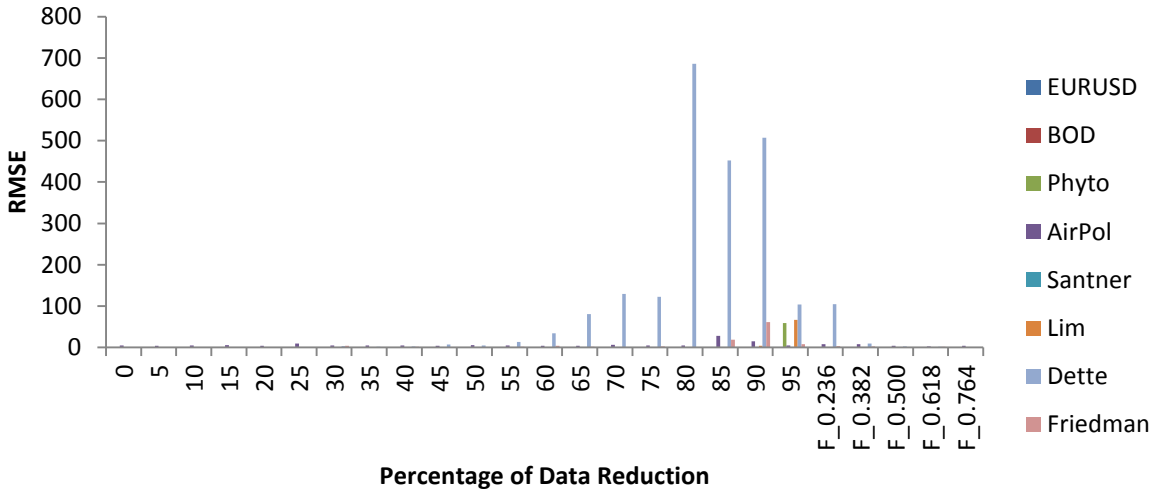

Figure-2. Overall RMSE value for RBFN training for each dataset.

Figure-2 showed that the groups from range 55 to 95 percent got spike up in RMSE values. While for the proposed group, only F_0.236 has the spike in RMSE values. Since the main concern for RBFN is about prediction, which indicate accuracy lead the time usage. It is necessary to exclude some data reduction size group to further investigate the performances of proposed method. The criterions to exclude a group in here are high score in RMSE value and time usage. Based on results in Figure-2, the groups of percentage range from 55 to 95 percent and F_0.236 must be excluded for further investigation. Then from Figure-1, the groups from range of 0 to 20 percent are excluded as they attained high time usage in RBFN training compared to other groups.

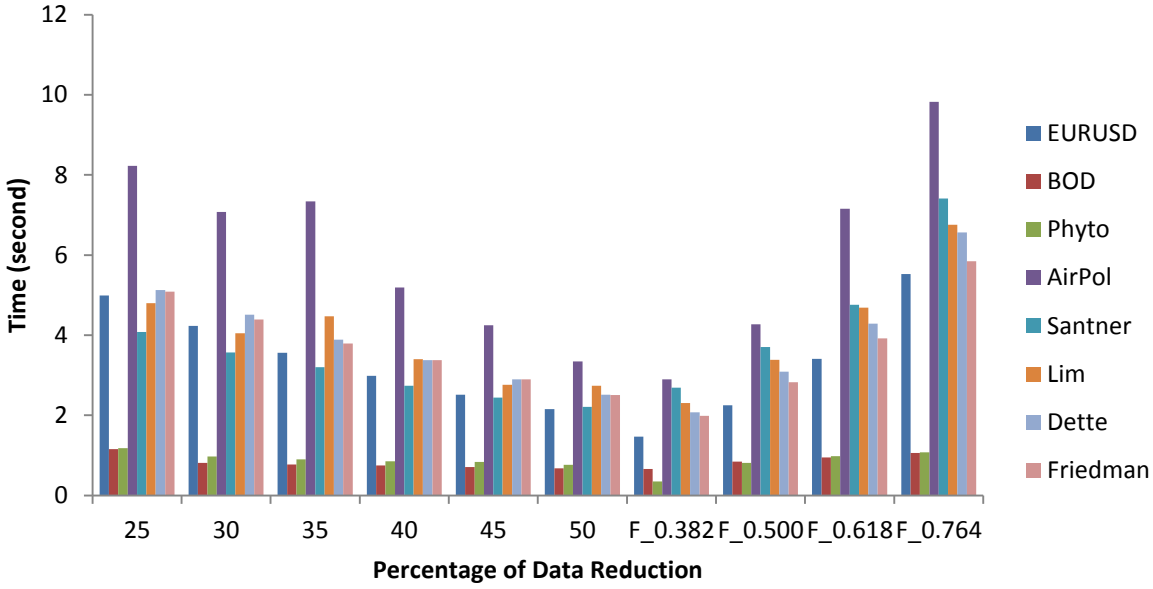

Figure-3. Comparison chart for time usage in RBFN training for selected data reduction size group.

Figure-3 displayed the selected groups for further investigation. Here, one can observed that F_0.382 outperformed the other groups in term of average time usage. On the contrary, F_0.764 obtained the highest average time usage of all groups.

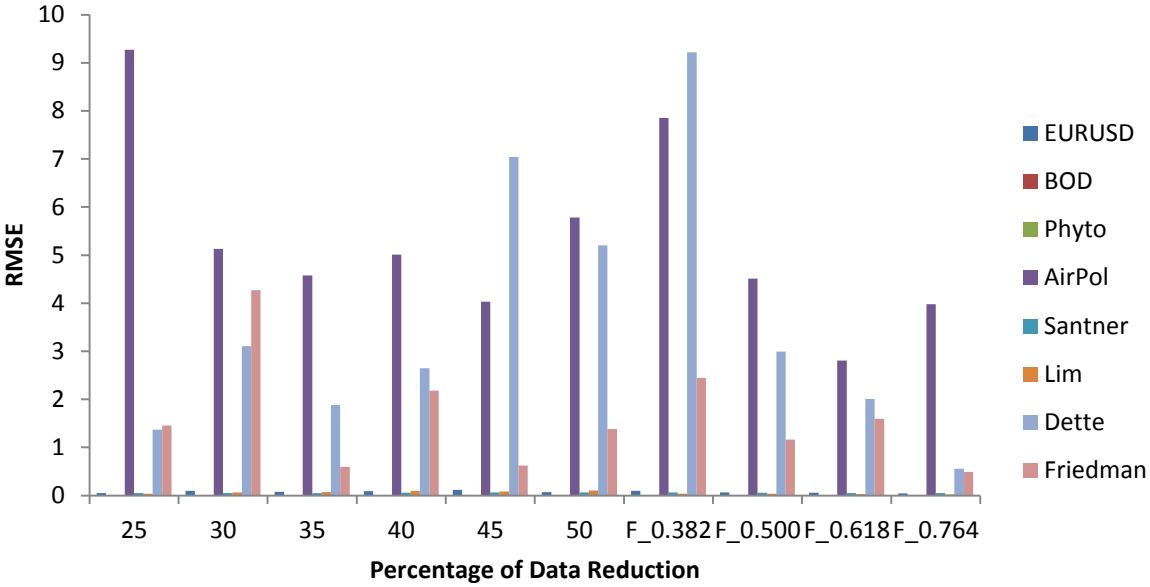

Figure-4. Comparison chart for RMSE in RBFN training for selected data reduction size group..

Moving to Figure-4, it is observed that many groups performed poorly on Air pollutant dataset and Dette function dataset training. Furthermore, the group from 25, 30 and 50 percent, and F_0.382 exceed the average RMSE value of 5.3 in Air Pollutant dataset. Meanwhile, the group from 45 and 50 percent with F_0.382 and F_0.764 exceed the average RMSE value of 3.6 in Datte function dataset. Hence based on Figure-3 and Figure-4 results, group from 25, 30 45 and 50 percent is exclude with F_0.382 and F_0.764 for further consideration in Fibonacci ratio assignment in proposed formula.

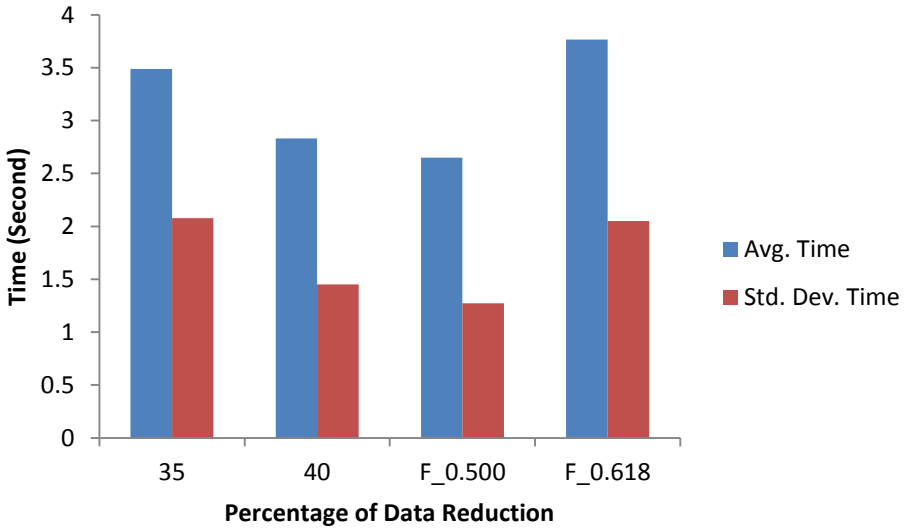

Figure-5. Comparison of average time usage and standard deviation of time usage for the selected data reduction group.

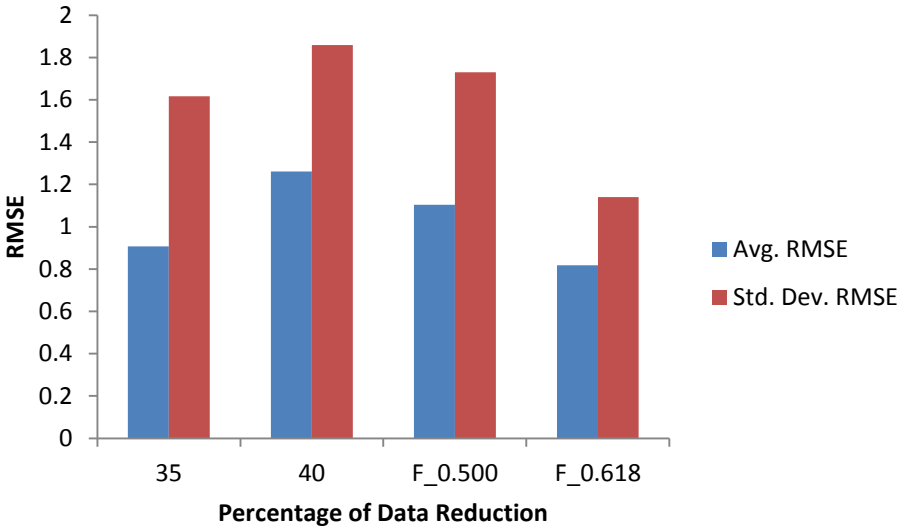

Figure-6. Comparison of average RMSE and standard deviation of RMSE for the selected data reduction group.

Next, Figure-5 showed that the difference in time usage for group 35 and 40 percent, and group F_0.500 and F_0.618 do not deviate significantly. However, Figure 6 showed that F_0.618 obtained the lowest standard deviation and average for RMSE values compared to group from 35 and 50 percent, and F_0.500.

The percentage of deviation of F_0.618 for average RMSE to group 35 and 40 percent are 9.85 percent and 35.12 percent, respectively. In addition, the percentage of deviation of F_0.618 to F_0.500 is 25.9 percent. This results shows that F_0.618 outperformed other groups in term of accuracy. Furthermore, F_0.618 also yield lowest standard deviation of RMSE value compared to other groups. Since priority is set in RMSE, hence F_0.618 with Fibonacci ratio of 0.618 is used in proposed formulation for dataset size reduction.

Finally, the proposed formula in equation (5) is rewritten into equation (13) by substitute the value of Fibonacci ratio of 0.618 into F_{ratio}.

$$S_{data} = 0.618N_{data} + \frac{N_{data}}{5A_{data}} \quad (13)$$

5. CONCLUSIONS

Experiments on four real-world problems and four literatures function has been simulated in this paper, where proposed dataset size reduction formula were applied on these case study for its effectiveness on obtaining suitable size for neural networks training dataset. Experiments were run using different percentage of size reduction in comparing to proposed formula. Results obtained were then compared for time usage in training RBFN network and RMSE for accuracies. The size reduction groups that obtained higher time usage and RMSE than their average values for time usage and RMSE, respectively, were excluded from further investigation. From the simulation, the Dette function dataset and air pollutant dataset were the most challenging datasets that give most groups score higher than average value. Finally, the remaining dataset reduction groups were compared for their average RMSE and standard deviation of RMSE, also the average and standard deviation for time usage in training RBFN.

Results from the experiments shows that the proposed dataset size reduction formula improves the RBFN training in term of accuracy and time usage. The finalized proposed formula found at the end of Section 4 is the optimal formula for RBFN network training. The proposed formula reduces the RMSE and time usage with an average of 26.2 percent and 69.1 percent, respectively. It is possible to improve the accuracy of the proposed formula by incorporating clustering method to choose the best value of points that represent the shape, based on the number of dataset given by the proposed formula, instead of just selecting the dataset points randomly. As conclusion, the proposed formula improves the RBFN as for learning speed and accuracy. For future work, it is noted that clustering method can perform self-organized selection of centers for selecting data points that represent the shape of the datasets; it would be interesting study if the proposed formula would be tested with clustering method using noisy training data to verify the efficiency of the dataset reduction number given by the proposed formula.

REFERENCES

Aik L. E. and Zainuddin Z. 2008. An Improved Fast Training Algorithm for RBF Networks Using Symmetry-Based Fuzzy C-Means Clustering Overview of Fuzzy C-Means Clustering Method. MATEMATIKA. 24(2): 141-148.

Albalate M. L. 2007. Data reduction techniques in classification processes. Retrieved from http://repositori.uji.es/xmlui/handle/10234/29617

Arteaga C. & Marrero I. 2013. Universal approximation by radial basis function networks of Delsarte translates. Neural Networks, 46, 299-305. https://doi.org/10.1016/j.neunet.2013.06.011

Bhattacharya S. and Kumar K. 2006. A computational exploration of the efficacy of Fibonacci Sequences in Technical analysis and trading. Annals of Economics and Finance. 1, 219-230.

Brenner R. and Maier-Paape S. 2016. Survey on log-normally distributed market-technical trend data. MDPI Open Access Journal. 4(3): 1-18. https://doi.org/10.3390/risks4030020

Broomhead D. S. and Lowe D. 1988. Radial Basis Functions, Multi-V ariable Functional Interpolation and Adaptive Networks. Royal Signals and Radar Establishment. 1-8. https://doi.org/10.1126/science.1179047

Brown C. 2012. Fibonacci Analysis. Fibonacci Analysis. https://doi.org/10.1002/9781118531587

Chih-Wei Hsu Chih-Chung Chang, & Lin, C. 2010. A Practical Guide to Support Vector Classification. Department of Computer Science, National Taiwan University. Retrieved from http://www.csie.ntu.edu.tw/cjlin/papers/guide/guide.pdf

Chouvatut V., Jindaluang W. and Boonchieng E. 2015. Training set size reduction in large dataset problems. 2015 International Computer Science and Engineering Conference (ICSEC). 1-5. https://doi.org/10.1109/ICSEC.2015.7401435

Dahiya K. 2014. Reducing Neural Network Training Data using Support Vectors. 0(2): 6-8.

Dasarathy B. 2002. Data mining tasks and methods: Classification: nearest-neighbor approaches. Handbook of Data Mining and Knowledge Discovery. Retrieved from http://portal.acm.org/citation.cfm?id=778257

Dasarathy B. V. 1991. A computational demand optimization aide for nearest-neighbor-based decision systems. In Conference Proceedings 1991 IEEE International Conference on Systems, Man, and Cybernetics (pp. 1777-1782). https://doi.org/10.1109/ICSMC.1991.169950

Demuth H. 2002. Matlab Neural Network. Networks, 24(1): 1-8. https://doi.org/10.1016/j.neunet.2005.10.002

Dette H. and Pepelyshev A. 2010. Generalized latin hypercube design for computer experiments. Technometrics. 52(4): 421-429. https://doi.org/10.1198/TECH.2010.09157

FOODY G. M., McCULLOCH M. B. and YATES W. B. 1995. The effect of training set size and composition on artificial neural network classification. International Journal of Remote Sensing. 16(9): 1707-1723. https://doi.org/10.1080/01431169508954507

Friedman J. H., Adaptive M. & Splines R. 1991. Multivariate Adaptive Regression Splines. The Annals of Statistics. 19(1): 1-67.

Fuangkhon P. and Tanprasert T. 2014. A training set reduction algorithm for feed-forward neural network using minimum boundary vector distance selection. In Proceedings - 2014 International Conference on Information Science, Electronics and Electrical Engineering, ISEEE 2014 (1: 71-78). https://doi.org/10.1109/InfoSEEE.2014.6948071

Gaucan V. 2011. How to use Fibonacci retracement to predict forex market. Scientific Papers, Economics.

Haykin S. 1994. Neural networks-A comprehensive foundation. New York: IEEE Press. Herrmann, M., Bauer, H.-U., & Der, R. https://doi.org/10.1017/S0269888998214044

Iqbal U., Ghazali R. and Shah H. 2018. Fibonacci polynomials based functional link neural network for classification tasks. Advances in Intelligent Systems and Computing (Vol. 700). https://doi.org/10.1007/978-3-319-72550-5_23

Jędrzejowicz J. and Jędrzejowicz P. 2016. Distance-based online classifiers. Expert Systems with Applications, 60, 249-257. https://doi.org/10.1016/j.eswa.2016.05.015

Kasemtaweechok C. 2015. Training Set Reduction using Geometric Median. 153-156.

Kavaklioglu K., Koseoglu M. F. & Caliskan O. 2018. International Journal of Heat and Mass Transfer Experimental investigation and radial basis function network modeling of direct evaporative cooling systems. International Journal of Heat and Mass Transfer, 126, 139-150. https://doi.org/10.1016/j.ijheatmasstransfer.2018.05.022

Kirsten M. and Wrobel S. 1998. Relational distance-based clustering. In Proceedings of the 8th international conference on Inductive logic programming, ILP-98, July 22-24 (pp. 261-270). Retrieved from http://books.google.com/books?hl=en&lr=&id=yZLhIJ4GFiAC&oi=fnd&pg=PA261&dq=Relational+Distance-Based+Clustering&ots=6lgnPbC7oN&sig=Qj8nqd93fjjVCt_GBAMabYVFFQQ

Kumar R. 2014. Magic of Fibonacci Sequence in Prediction of Stock Behavior. International Journal of Computer Applications. 93(11): 36-40. https://doi.org/10.5120/16262-5926

L G. C. and Perez R. 1998. A Data Reduction Method To Train, Test, and Validate Neural Networks 1 Gerardo Colmenares L' and Rafael Perez, 277-280.

Lei Y., Ding L. and Zhang W. 2015. Generalization Performance of Radial Basis Function Networks. Neural Networks and Learning Systems, IEEE Transactions On, 26(3): 551-564. https://doi.org/10.1109/TNNLS.2014.2320280

Lim Y. B., Sacks J., Studden W. J. and Welch W. J. 2002. Design and analysis of computer experiments when the output is highly correlated over the input space. Canadian Journal of Statistics. 30(1): 109-126. https://doi.org/10.2307/3315868

Lin S. 2016. Linear and nonlinear approximation of spherical radial basis function networks, 35, 86-101. https://doi.org/10.1016/j.jco.2016.02.003

Liu H., Zhang X., Zhang X. and Cui Y. 2017. Self-adapted mixture distance measure for clustering uncertain data. Knowledge-Based Systems. 126, 33-47. https://doi.org/10.1016/j.knosys.2017.04.002

Loh Wei-Yin and Yu-Shan Shih. (1997). Split Selection Methods for Classification Trees. Statistica Sinica. 7(4): 815-840.

Lozano M., Sánchez J. and Pla F. 2003. Reducing training sets by NCN-based exploratory procedures. Pattern Recognition and Image Analysis, 453-461. Retrieved from http://link.springer.com/chapter/10.1007/978-3-540-44871-6_53

MacLean G. A. 2005. Fibonacci and Gann Applications in Financial Markets: Practical Applications of Natural and Synthetic Ratios in Technical Analysis. Wiley trading series.

Majdisova Z. and Skala V. 2017. Radial basis function approximations : comparison and applications. Applied Mathematical Modelling, 51, 728-743. https://doi.org/10.1016/j.apm.2017.07.033

Mohsen H., Kurban H., Jenne M. & Dalkilic M. 2014. A new set of Random Forests with varying dynamic data reduction and voting techniques. DSAA 2014 - Proceedings of the 2014 IEEE International Conference on Data Science and Advanced Analytics, 399-405. https://doi.org/10.1109/DSAA.2014.7058103

Naka M., Ino K. and Kohmoto M. 2005. Critical level statistics of the Fibonacci model. Physical Review B - Condensed Matter and Materials Physics, 71(24). https://doi.org/10.1103/PhysRevB.71.245120

Ougiaroglou S., Diamantaras K. I. and Evangelidis G. 2017. Exploring the effect of data reduction on Neural Network and Support Vector Machine classification. Neurocomputing, 280, 101-110. https://doi.org/10.1016/j.neucom.2017.08.076

Ougiaroglou S. and Evangelidis G. 2012. Efficient dataset size reduction by finding homogeneous clusters. Proceedings of the Fifth Balkan Conference in Informatics, (i): 168-173. https://doi.org/10.1145/2371316.2371349.

Rico-Juan J. R. and Iñesta J. M. 2012. New rank methods for reducing the size of the training set using the nearest neighbor rule. Pattern Recognition Letters. 33(5): 654-660. https://doi.org/10.1016/j.patrec.2011.07.019.

Roy P. P., Leonard J. T., and Roy K. 2008. Exploring the impact of size of training sets for the development of predictive QSAR models. Chemometrics and Intelligent Laboratory Systems, 90(1): 31-42. https://doi.org/10.1016/j.chemolab.2007.07.004

Sánchez J. S. 2004. High training set size reduction by space partitioning and prototype abstraction. Pattern Recognition. 37(7): 1561-1564. https://doi.org/10.1016/j.patcog.2003.12.012

Santner T. J., Williams B. J., and Notz W. I. 2003. The Design and Analysis of Computer Experiments (1st ed.). Springer-Verlag New York. https://doi.org/10.1007/978-1-4757-3799-8

Sebtosheikh M. A. and Salehi A. 2015. Lithology prediction by support vector classifiers using inverted seismic attributes data and petrophysical logs as a new approach and investigation of training data set size effect on its performance in a heterogeneous carbonate reservoir. Journal of Petroleum Science and Engineering, 134, 143-149. https://doi.org/10.1016/j.petrol.2015.08.001

Shayegan M. A. and Aghabozorgi S. 2014. A new dataset size reduction approach for PCA-based classification in OCR application. Mathematical Problems in Engineering, 2014. https://doi.org/10.1155/2014/537428

Smolik M., Skala V. and Majdisova Z. 2018. Advances in Engineering Software Vector fi eld radial basis function approximation ☆. Advances in Engineering Software. 123(17): 117-129. https://doi.org/10.1016/j.advengsoft.2018.06.013

Wang S., Li Z., Liu C., Zhang X. and Zhang H. 2014. Training data reduction to speed up SVM training. Applied Intelligence. 41(2): 405-420. https://doi.org/10.1007/s10489-014-0524-2

XM Global L. 2018. XM Metatrader 4. Retrieved July 20, 2018, from https://www.xm.com/mt4

Yousef W. A. and Kundu S. 2014. Learning algorithms may perform worse with increasing training set size: Algorithm-data incompatibility. Computational Statistics and Data Analysis. 74, 181-197. https://doi.org/10.1016/j.csda.2013.05.021

Zhou Z.-H., Wei D., Li G. and Dai H. 2004. On the size of training set and the benefit from ensemble. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). (Vol. 3056). https://doi.org/10.1007/b97861.