Journal of Fundamental and Applied Sciences

ISSN 1112-9867

Available online at http://www.jfas.info

Research Article

Special Issue

INPUT SIGNIFICANCE ANALYSIS: FEATURE RANKING THROUGH SYNAPTIC WEIGHTS MANIPULATION FOR ANNS-BASED CLASSIFIERS

R. Hassan*[1], I. F. T. Al-Shaikhli[1] and S. Ahmad[2]

[1]Department of Computer Science, Kulliyyah of Information and Communication Technology, International Islamic University Malaysia, 50728 Kuala Lumpur, Malaysia

[2]Department of Mechatronics Engineering, Kulliyyah of Engineering, International Islamic University Malaysia, 50728 Kuala Lumpur, Malaysia

Published online: 05 October 2017

ABSTRACT

Due to the ANNs architecture, the ISA methods that can manipulate synaptic weights selected are Connection Weights (CW) and Garson's Algorithm (GA). The ANNs-based classifiers that can provide such manipulation are Multi-Layer Perceptron (MLP) and Evolving Fuzzy Neural Networks (EFuNNs). The goals for this work are firstly to identify which of the two classifiers works best with the filtered/ranked data, secondly is to test the FR method by using a selected dataset taken from the UCI Machine Learning Repository and in an online environment and lastly to attest the FR results by using another selected dataset taken from the same source and in the same environment. There are three groups of experiments conducted to accomplish these goals. The results are promising when FR is applied, some efficiency and accuracy are noticeable compared to the original data.

Keywords: artificial neural networks, input significance analysis; feature selection; feature ranking; connection weights; Garson's algorithm; multi-layer perceptron; evolving fuzzy neural networks**.**

Author Correspondence, e-mail: hrai@iium.edu.my

doi: http://dx.doi.org/10.4314/jfas.v9i4s.37

1. INTRODUCTION

A long-lived and one of the popular techniques used to solve classification problems is Artificial Neural Networks (ANNs). It is also known as a powerful modeling technique that is used in many other applications such as speech recognition, data mining, machine control and financial series forecasting. Nonetheless, it is also frequently viewed as a black box due to the vague or very little description over the contributions of the independent variables/input data/input features in the prediction process [1]. Therefore, ANNs is not able to provide a good understanding and interpretation of itself, based on how the results were reached and whether there exist input data/features that play greater or more important role in contributing such results.

In relation to the classification problem, due to the information that can grow over time and eventually become huge, it is important for classifiers to be integrated with the method(s) that can understand the role of each input data/features in the datasetbefore processing takes place. This can help to identify redundant and irrelevant input data/features which when removed, can speed up the processing time.

Over the years, as one of the attempts to shed some lights inside the black box, Input Significance Analysis (ISA) is a research interest that focuses on finding the role of input data/features that could influence the results. From this research interest, many associated methods have been created.

From one of the earlier works, ISA is proven to be able to provide an audit trail that can help in explaining on how the system reached its decisions or conclusions [2]. In other words, the whole ISA processes can lead to the identification of the best data that is relevant to the problem in hand. With regards to ANNs, it can reduce the network processing time and that can lead to a faster training, reducing the size of the network so that its structure will be optimal and possibly can produce better results [3]. Also in the context of ANNs, ISA can be defined as the following:

1. The methods applied in a Multi-Layer Perceptron (MLP) to establish the significance or rank of each of the input data/features (MLP) [4].
2. The relation contribution estimation procedures of each of input data/features [5].

Additionally, the ISA processes may also resemble variable or feature selection (FS). The

prominent researchers in this area, in [6] described the data that have no redundancy and irrelevancy is the best representation of data. Other outstanding researchers, in [7] described FS as when the irrelevant features are removed, the measure of accuracy, consistency, information, distance or dependence of the remaining features will improve. Another simple definition of FS is to reduce the number of variables or input features or input neurons[8]. With regards to classification, FS allows that while the classifier's structure is as simple as possible, the classifier still can obtain accurate prediction [7]. The following are the objectives of FS [6]:

1. The prediction performance of the predictor/classifier can be improved.
2. The processes involved in generating the data can be better understood.

Overall, the following are the results of FS [7]:

1. The algorithm can learn faster because of fewer data to be processed.
2. The accuracy is optimal, thus allowing better generalization from the data for the classifier.
3. Results that is simple and easy to understand.
4. Lesser features, hence no need to run the redundant or irrelevant features.

In both ISA and FS, there can be a process or step called feature ranking (FR). FR is referring to a ranked features list, ordered according to evaluation measures [7].

In this research, the ISA techniques called Connection Weights (CW) and Garson's Algorithm (GA) are the evaluation measures selected. Once the features in the dataset are ranked according to these evaluation measures, FS is then applied to produce a smaller version of the dataset, so that the FS results stated above can be achieved. In this work, only FR will be applied first.

This approach has also been adopted in a recent and similar work for histology image classification by which the subset selection and also the FR, made up the applied FS framework. The results showed that even the set of descriptors (the specific name for the dataset in this work) used is only 0.656% from the initial set, the framework could reach 90.5% accuracy [9].

In another work, the FR is used in combination with Principal Component Analysis (PCA) to solve the Curse of Dimensionality problem. The Curse of Dimensionality is referring to a

large or huge dataset and the problem, which is a popular problem in classification tasks. For the classifier itself is when the dataset is too big, it can cause the classifier to produce less accurate results. Though it is not mentioned specifically as FS, this work reduced the dataset based on the combination of PCA and FR and found that the classification accuracy and computation cost are not compromised [10].

The next example is FR in an incremental FS. The incremental FS is referring to the changing environment that this work performed on, in which the latest instances contains more features than the initial features or original data. The FR applied is based on the introduction of new features information, in specific the pre and post information. The FR then combines this information, even when the new features' instances is small in number. The results showed that when FR is applied to the initial features or original dataset, the FS performed on the new and larger-dimensional (due to the addition of the latest instances) dataset is improved [11].

Therefore, it can be concluded that FR is an important step prior to FS and works even in a changing environment such as an online environment where the data is accumulating and changing at all times.

For the works focused solely on FR, recent works showed that it is applied in various applications, and mainly to solve the big/large/high-dimensionality data that can compromise the results.

For example, a work on regression problem that applied a new FR method and stated that FR may be used as a complementary step to reduce the size of high-dimensionality of real-world data. The results showed that in term of effectiveness, the proposed FR method is a valid alternative to existing methods [12].

Another work is the use of FR to solve a large number of features problem in big data field. FR is achieved by using Hadoop implementation and MapReduce is used to calculate the distributed and parallel computation of information gain, used as the evaluation measure in FR. A minor contribution of this work is the computations that are distributed and parallelized on a cluster can achieve a significance speedup [13].

There are also works on FR that use special FR names, based on their specific function. For example, in a 3D objects retrieval application, a manifold ranking is used as the FR method. This type of ranking is referred to a technique to re-arrange the retrieved results [14].

Then there is an ensemble FR that is created to produce a more robust ranking. This ranking comprises of 8 feature ranking algorithms that were combined by using the Schulze aggregation method [15].

In a robust speaker recognition application, feature warping is used together with FR to create an improved ranking-based feature enhancement called Revised Feature Warping (RFW). The function of feature warping is to decrease the mismatch between the training and testing environment, which is commonly used in an automatic speaker or speech recognition field [16].

Finally, in a visual data mining application, a Ranked Ordered Feature List (ROFL) is used to identify the least dominant features. These identified features are then removed in order to improve the classification accuracy [17].

While many classifiers have no problems in tackling classification tasks, there are questions about the datasets themselves. Does the dataset contain just helpful data? Does the dataset contain no repetitive data? So, these are the issues that can be tackled in the classification problem, and when they are being addressed, improved results can be generated. From the classification algorithms point of view, the disadvantages of having these undesirable features are [7]:

1. When there are too many input data, that means there will be more instances needed and this can cause a classification algorithm taking a longer time in learning the extra data.
2. The data that are redundant or irrelevant can misguide the learning algorithms and can cause overfitting of data.

Historically, ANNs birth can be traced from the days of Connectionist Systems (CSs). CSs is a type of Computational Intelligence (CI) area. CSs are referring to the systems that are based on layers of neurons in the human brain. This inspiration came from one of the significant element of a human brain, which is multi-tasking without much effort. Hence, it is also referred to as Neurally Inspired Models (NIMs). As for CI, it is an area within Computer Science that aims to solve the problems of real-world that are complex in nature whereby the conventional approaches are inefficient. Generally, CI is referring to a set of nature-inspired computational methodologies and approaches.

Other than that, CSs are also called Parallel Distributed Processing (PDP) due to distributed

processing among all neuron layers. Neural Networks (NNs) are one of the earliest connectionist models, followed by its enhanced version called ANNs. Haykin stated that ANNs mimics the brain in two ways [18]:

1. Learning the process of a network is the method to acquire knowledge.
2. The interconnection strengths between neurons are representing the synaptic weights or weights that are used to store data.

The second way mentioned above is used as the basis of selection for ISA techniques. The synaptic weights or weights values can be manipulated. For example, based on the highest weight to the lowest one, a feature ranking table can be produced.

There are two ANNs-based classifiers selected for this work:

1. Multi-Layer Perceptron (MLP)
2. Evolving Fuzzy Neural Networks (EFuNNs)

The MLP is an example of ANNs that is used extensively in many applications. It can solve some of the different problems such as the ones that come from pattern recognition and interpolation applications [19]. The MLP architecture is referring to multiple layers that have simple, two-state, sigmoid processing elements or neurons. The interaction between them is achieved through weighted connections [20]. Basically, it adds one or more layers of neurons to the basic perceptron architecture. The principle weakness of MLP is that it could only solve problems that are linearly separable [19].

EFuNNs is the advanced concept of ANNs that combines ANNs with Fuzzy Logic (FL) and come from the family of Evolving Connectionist Systems (ECoSs). Basically, there are two types of Fuzzy Neural Networks (FuNNs). The one where interpretations of fuzzy rules define the connectionist structures is the type to which EFuNNs belongs. EFuNNs can be detailed out as first, a FuNNs where the implementation of fuzzy rules and fuzzy inference is handled by its connectionist structures and second as an EFuNNs itself, the FuNNs that change over time based on ECoS principles [21]. In simple terms, it is characterized by embedded fuzzy logic elements [22].

There are two knowledge manipulation techniques for EFuNNs. The first technique is the combination of rule insertion and extraction. The fuzzy or exact rules can be inserted or extracted from anytime/phase of the learning process. The second technique is the

rulesaggregation. This technique allows several rule nodes to be merged into one [21]. The main advantage of EFuNNs is its knowledge manipulation that causes it to be able to explain what it has learned in an understandable form. Additionally, the investigation on the role of ISA has been highlighted as one of the future directions of ECoSs in a particular paper about a decade of ECoSs[4].

For ISA techniques, the two selected techniques are specifically chosen because they can manipulate the synaptic weights and they are:

1. Connection Weights (CW)
2. Garson's Algorithm (GA)

CW and GA have been applied in recent works [23-25]. Specifically for CW, it has been applied in [26-27]. For GA, the works are in [28-31].

There are contradicting findings. A work that accurately compares the methods for quantifying variable significance or rank in ANNs reported that CW performed the best. The correctly ranked significance values of all predictor variables were consistently identified by CW. While for GA, it has been reported as the poorest performed technique in the same work mentioned above [1].

Another recent work applied CW to evaluate the factors responsible for industrial energy consumption CW is applied in an MLP network.Whereby, the activity, structure and intensity served as input parameters and energy consumed served as the output parameter. The results showed that CW has successfully identified that the intensity factor should be looked on to first as a means of saving energy, followed by activity and structure factors [27].

As for GA, apart from the work mentioned earlier, it was also found to perform poorer in the work of [30]. When a connection weight between the input-hidden layers is negative and meets with a connection weight between the hidden-output that is positive, they could cancel out each other, and the final output in the result can be not significantly important. This flaw can be linked to the deficiency of GA, i.e. the variable contributions were calculated using absolute connection weights that exclude the counteracting connection between weights linking input and output neurons [1].

In another work where GA was found to have mixed performance, the GA was used in combination with correlation and sensitivity analysis to identify the most important

parameters in deciding the PV power amount. When tested individually, the performance of each approach was unsatisfactory. Measuring metric for the input variable importance was devised through a combination of all as a means to solve this problem, and the results showed improved forecast accuracy and subsequently proved that this selection procedure is effective [28].

Nevertheless, especially in ecological literature, the most common method used is GA [1]. Therefore, there exist some of the advantages reported in that ecological literature. For example, a work that reported that GA has benefits when used in the model-building stage [30].

There is another advantage of GA that was reported in the work in which GA was found to be one of the most sensitive techniques, despite not belonging to the same group of other different ANNs methods that seemed to give similar results concerning the order of significance. The diverse computation of these other ANNs methods caused their variation in sensitivity and stability. GA is found to be able to distinctly separate the environment properties into minor and major contributors [32]. Additionally, as mentioned earlier, in the original paper of Garson's algorithms stated that it is best used for causal or sensitive analysis applications [2].

GA was also found to be suitable to be used with Backpropagation Network (BPN). GA was applied in a backpropagation network (BPN) to capture the significance of interactions among the input variables. By using BPN together with GA, all input variables are allowed to vary simultaneously which subsequently creates a multi-parameter sensitivity analysis [31].

Finally, it is worth to highlight that there are other Machine Learning (ML) classifiers that applying the same method as in this work. For example, a work that combines correlation criteria as the feature selection method and Support Vector Machine (SVM) as the classification part [33]. Another example of work uses Genetic Algorithms (GA) as the feature selection technique and SVM as one of the classifiers used [34].

2. MATERIAL AND METHOD

This section will describe the experimental setup used by this work.

2.1. Working Environment

The basic laptop is used as the machine to carry out the classification experiments by using the MATLAB software.

2.2. Data

There are two datasets used. The first one is called gas sensor array drift (or shortly as gas sensors) [35-36]. The second one is called steel plate's faults dataset (or shortly as steel) [37] and both are from UCI Machine Learning Repository [38].

For the gas sensors dataset, originally it came from a large dataset that contains 13910 measurements from 16 chemical sensors. These 16 chemical sensors are exposed to 6 different gassesat different concentration levels. There are ten batches (ten data collections), organized into 128 independent variables and 6 dependent variables. The 6 dependent variables are [35-36]:

1. Ethanol
2. Ethylene
3. Ammonia
4. Acetaldehyde
5. Acetone
6. Toluene

The ethanol classification data is the only data used by this work. After searching this data in all ten batches, a dataset with 2505 instances or rows are identified and with 128 input features and 1 output feature (ethanol classification).

For the steel dataset, there are 1941 instances with 27 independent variables and 7 dependent variables which makes the dataset multivariate. The 7 dependent variables (7 types of steel plates faults) are [37]:

1. Pastry
2. Z_Scratch
3. K_Scratch
4. Stains
5. Dirtiness
6. Bumps

7. Other_Faults

2.3. Validation Methods

Four validation methods are used in this research and they are:

1. Memory recall
2. Random subsampling
3. K-Fold cross-validation
4. Holdout

For the memory recall validation method as the name suggests, it means to test the classifier's recall capability, i.e. 100% dataset is used for both training and testing. This validation method is applied in both gas sensors and steel datasets.

The random subsampling validation method is only applied to gas sensors dataset. The dataset's 2505 rows are separated into 2405 rows for the training set, and the remaining 100 random rows for the testing set. This step is repeated another 2 more times.Therefore, the overall (averaged) calculations of all measurements will be used.

For the K-Fold cross-validation method, it is also only applied to gas sensors dataset. The 2505 rows of the datasetare divided into 1670 rows of the training set and the remaining 835 selected rows as testing set. This step is repeated three times; hence, the overall (averaged) calculations of all measurements will be used. The K-Fold cross-validation method is completed when all datum in the dataset become both trained at one point and tested at another point in time.

Lastly, for the holdout validation method, it is applied to both datasets. For the gas sensors,dataset selected 1755 rows of the dataset are for the training set and the 750 rows left are for the testing set. For the steel dataset, selected 1294 rows of the dataset are selected for the training set, and the 647 rows left are for testing set.

2.4. Weights Initialization

To be able to calculate the importance value of a feature in the dataset before being applied in the classifier, synaptic weights initialization values were obtained through Nguyen-Widrow layer initialization function called INITNW in MATLAB. The Nguyen-Widrow initial weights distribution is originally designed to improve the learning speed of 2-layer NNs [39]. This improvement is achievable when it generates the initial values for the weights and bias of

a layer in order to make the active regions of the layers neurons disseminated approximately uniformly over the input space [40-41].

This weight initialization function has been specifically chosen and applied in the works of [42-43]. It has been claimed as the best function generating weights and biases that increase the speed of the training [44]. It is also claimed that a better starting point to the algorithm used can be created by using the generated weights and biases for input to hidden layer nodes, obtained from this function [45].

2.5. CW

The formula is referred below [1]. First, the product between the raw input-hidden and hidden-output connection weights of between each input neuron/feature and output neuron/feature are calculated. After that, the products across all hidden neurons/features as summed up [46].

$$\text{Input}_X = \sum_{Y=A}^{E} \text{Hidden}_{XY} \quad (1)$$

2.6. GA

The formula is referred below [1]. First, the partitioning into components is applied to the hidden-output connection weights. After that, each input neuron/feature that uses absolute values of connection weights is then linked to these components [46].

$$\text{Input}_X = \sum_{Y=A}^{E} \frac{|\text{Hidden}_{XY}|}{\sum_{Z=1}^{5} |\text{Hidden}_{ZY}|} \quad (2)$$

2.7. Steps for CW and GA

The following Fig. 1 shows the steps for CW and GA [1].

Input-Hidden Connection Weights

X \ Y	Hidden A	Hidden B	Hidden C	Hidden D	Hidden E
Input 1	-0.93	-1.49	0.37	-0.91	0.37
Input 2	-0.57	-1.96	-0.14	1.18	1.26
Input 3	-0.85	1.74	-1.86	-0.05	0.10
Input 4	0.25	-3.01	-0.99	-1.34	-1.65
Input 5	-0.82	0.09	0.86	-0.41	-0.05

×

Hidden-Output Connection Weights

	Hidden A	Hidden B	Hidden C	Hidden D	Hidden E
Output	-1.75	-1.08	-1.13	-2.90	3.37

Connection Weight Products

X \ Y	Hidden A	Hidden B	Hidden C	Hidden D	Hidden E
Input 1	1.63	1.62	-0.42	2.64	1.24
Input 2	1.00	2.12	0.16	-3.43	4.25
Input 3	1.48	-1.89	2.10	0.14	0.34
Input 4	-0.43	3.26	1.12	3.90	-5.57
Input 5	1.43	-0.09	-0.97	1.18	-0.16

$$Input_X = \sum_{Y=A}^{E} Hidden_{XY}$$

$$Input_X = \sum_{Y=A}^{E} \frac{|Hidden_{XY}|}{\sum_{Z=1}^{5} |Hidden_{ZY}|}$$

	Importance	Rank
Input 1	6.71	1
Input 2	4.10	2
Input 3	2.18	3
Input 4	2.28	4
Input 5	1.38	5

Connection Weight Approach

	Importance	Rank
Input 1	0.88	4
Input 2	1.11	2
Input 3	0.94	3
Input 4	1.50	1
Input 5	0.57	5

Garson's Algorithm

Fig.1. Steps for CW and GA

2.8. Performance Measurements

There are three principal dimensions of FS: search strategy, evaluation measure and feature generation scheme [7]. The following Fig. 2 illustrates these dimensions.

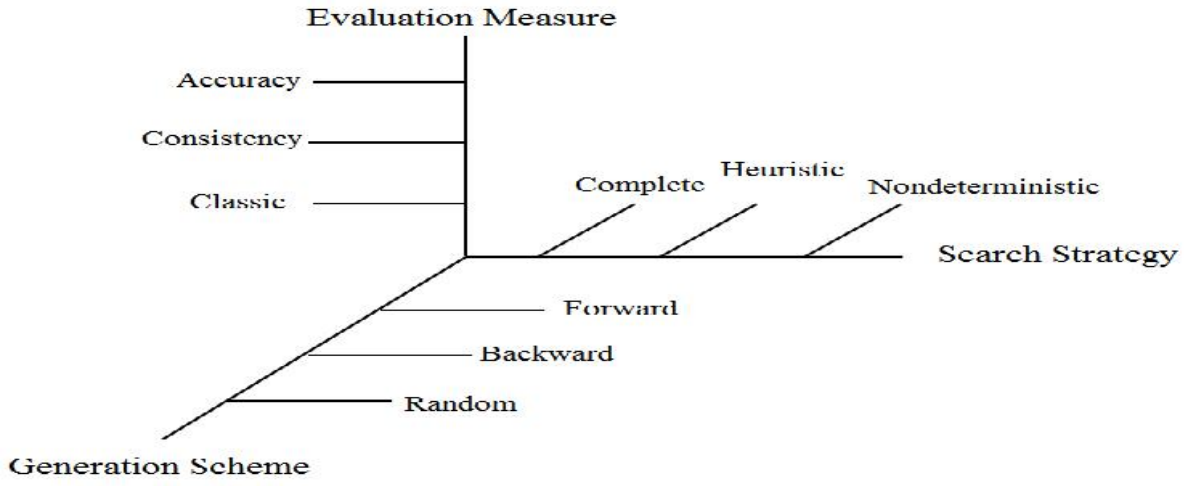

Fig.2. Three principal dimensions of FS

This research is interested in the performance of the classifiers. Therefore, evaluation measure is selected. The measurements selected are:

1. Elapsed training time, etrt(seconds):This measurement is used to observe the network processing time during training.

2. Overall elapsed training time, otrt(seconds):This measurement refers to the averaged elapsed training time. The formula is:

$$\text{otrt} = \frac{\text{etrt}_1 + \text{etrt}_2 + \text{etrt}_n}{n} \quad (3)$$

wheren = the count number of training time.

3. Elapsed testing time, ett(seconds):This measurement is used to observe the network processing time during testing.

4. Overall elapsed testing time, ott(seconds):This measurement refers to the averaged elapsed testing time. The formula is:

$$\text{ott} = \frac{\text{ett}_1 + \text{ett}_2 + \text{ett}_n}{n} \quad (4)$$

wheren = the count number of testing time.

5. Root-mean-squared-error, RMSE:This is an estimator and is one of many ways to quantify the difference between values implied by an estimator and the true value of the quantity being estimated. The lower the RMSE, the better fitting is the model. The formula is:

$$\text{RMSE}(f) = (\text{sqrt}(\frac{1}{m} + \sum_{i=1} m\,(f(x_i) - y_i)^2)) \quad (5)$$

wherem = the number of test examples, $f(x_i)$ = the classifier's probabilistic output on x_i and y_i = the actual label.

6. Overall root-mean-squared-error, oRMSE:This measurement refers to the averaged RMSE. The formula is:

$$\text{oRMSE} = \frac{\text{RMSE}_1 + \text{RMSE}_2 + \cdots \text{RMSE}_n}{n} \quad (6)$$

wheren = the count number of RMSE.

7. Resubstitutionerror, re (percentage): This refers to the error rate of the training data. It is an estimator of error based on the differences between the predicted values of a trained model and the observed values in the training set. This measurement is likely not a good indicator of future performance because it does not process the unseen data. The formula is:

$$re = \text{predicted values } (data_X) - \text{observed values } (data_X) \quad (7)$$

where$data_X$ = refers to the whole dataset.

8. Error rate, er(percentage): This measurement refers to the error rate from the training and testing of different data. It means that for testing, unseen data is processed. The formula is:

$$re = \text{predicted values } (data_X) - \text{observed values } (data_Y) \quad (8)$$

where$data_X$ = refers to say 30% of the dataset and $data_Y$ = refers to the remaining 70% of the dataset

9. Overall error rate, oer(percentage): Overall error rate simply means the averaged error rate. The formula is:

$$oer = \frac{er_1 + er_2 + er_n}{n} \quad (9)$$

where$n$ = the count number of er.

10. Created nodes, cn:This measurement refers to the number of nodes created initially at the start of the classification process.

11. Pruned nodes, pn:This measurement refers to the number of nodes pruned or optimized towards the end of the classification process, in order to ensure optimal functions of the classifier for the next classification tasks.

$$pn = f(cn) \quad (10)$$

where$f(cn)$ = function that perform optimization on created node, cn.

12. Leftover (rule) nodes, ln:This measurement refers to the number of nodes left after the classification process has ended.

$$ln = cn - pn \quad (11)$$

2.9. Experiments

There are three groups of experiments conducted.

1. Group 1: To choose a classifier

This group of experiments applied selected ISA techniques in the dataset. The new, filtered dataset then known as an ISA-filtered dataset. ISA-filtered dataset simply means that the original dataset has been manipulated in such a way that the order of the columns of data (features) has been re-arranged according to importance value. This group of experiments aimed to capture the effect(s) of using the ISA-filtered dataset in their end results.

2. Group 2: To test the FR method

This group of experiments will be conducted in an online environment or self-tuning EFuNNs. These experiments are needed to prove that by processing the data in the online environment or self-tuning EFuNNs, it can give better results than the normal mode of EFuNNs.

3. Group 3: To test FR method by using different dataset

This group of experiments aimed to attest the results obtained in Group 2 above, by using different dataset, in the same online environment or self-tuning mode.

3. RESULTS AND DISCUSSION

3.1. Group 1

There are three validation methods applied in this group of experiments and they are memory recall, random subsampling and K-Fold cross-validation. The dataset used is gas sensors, and the ISA-filtered data are CW- and GA-ranked data.

Table 1. Classification experiments using memory recall validation method

Classifiers	Measurements	CW-Ranked Data	GA-Ranked Data
MLP	Elapsed Training Time (Seconds)	1.98	2.715
	Elapsed Testing Time (Seconds)	0.06	0.067
	Root-mean-squared-error (RMSE)	33.23	8.04
	Resubstitution Error (Percentage)	95.37	85.51
EFuNNs	Elapsed Training Time (Seconds)	24.53	25.06
	Elapsed Testing Time (Seconds)	4.95	5.08
	Root-mean-squared-error (RMSE)	31.32	27.08
	Resubstitution Error (Percentage)	78.00	81.88

For the two classifiers, it is found that CW-ranked data took less training and testing time than GA-ranked data. For the RMSE measurement, GA-ranked data was found to make both classifiers better fitting models compared to CW-ranked data.

For the last measurement, specifically for MLP, GA-ranked data was found to be better than CW-ranked data and for EFuNNs, the latter was found to be the best between the two ranked data.

Table 2. Classification experiments using random subsampling validation method

Classifiers	Measurements	CW-Ranked Data	GA-Ranked Data
MLP	Overall Elapsed Training Time (Seconds)	6.23	3.77
	Overall Elapsed Testing Time (Seconds)	0.05	0.06
	Overall Root-mean-squared-error (RMSE)	33.46	12.81
	Overall Error Rate (Percentage)	90.33	91.67
EFuNNs	Overall Elapsed Training Time (Seconds)	25.37	24.24
	Overall Elapsed Testing Time (Seconds)	0.25	0.22
	Overall Root-mean-squared-error (RMSE)	22.018	17.66
	Overall Error Rate (Percentage)	84	93

For the two classifiers, it is found that GA-ranked data took less training time and making both classifiers better fitting models than CW-ranked data. On the other hand, CW-ranked data was found to have fewer classification errors than GA-ranked data.

For the testing measurement, specifically for MLP, CW-ranked data took less time compared GA-ranked data. ForEFuNNs, the latter was found to be the best between the two ranked data.

Table 3. Classification experiments using K-fold cross-validation method

Classifiers	Measurements	CW-Ranked Data	GA-Ranked Data
MLP	Overall Elapsed Training Time (Seconds)	5.5	2.31
	Overall Elapsed Testing Time (Seconds)	0.08	0.05
	Overall RMSE (Root-mean-squared-error)	28.81	30.21
	Overall Error Rate (Percentage)	91.98	90.26
EFuNNs	Overall Elapsed Training Time (Seconds)	42.08	25.04
	Overall Elapsed Testing Time (Seconds)	2.408	1.57
	Overall RMSE (Root-mean-squared-error)	36.458	28.57
	Overall Error Rate (Percentage)	85.87	85.59

For the two classifiers, it is found that GA-ranked data took less training and testing time and have fewer classification errors than CW-ranked data.

For RMSE measurement, specifically for MLP, CW-ranked data was found to make MLP [47] a better fitting model compared GA-ranked data. ForEFuNNs, the latter was found to be the best between the two ranked data.

For overall conclusion in this Group 1 experiments, GA-ranked data was found to perform better as an ISA-filtered data. Between the two classifiers, EFuNNs was found to work well with GA-ranked data. Therefore, EFuNNs is the only classifier that will be used in the next two groups of experiments.

3.2. Group 2

There are two validation methods applied in this group of experiments; they are memory recall and holdout. The classifier selected is EFuNNs and the data used will be original gas sensors dataset, CW- and GA-ranked gas sensors data.

For the measurements involving nodes, only created nodes and rule nodes will be described. The created nodes measurement will give the initial number of nodes at the point of creation. The smaller the number of nodes indicates that the classifier's network does not need to be larger than necessary to start running a classification process.

For the rule nodes measurement, it will give the number of nodes optimized (reduced) at the end of the classification process. The larger the number means the next classification process will start with a smaller number of nodes, and most probably the training and testing time also will be reduced compared to the first classification process.

Table 4. Classification experiments using memory recall validation method

Measurements	Original Data	CW-Ranked Data	GA-Ranked Data
Elapsed Training Time (Seconds)	447.36	436.1	442.4
Elapsed Testing Time (Seconds)	3.13	3.25	2.88
Created Nodes	282	344	304
Pruned Nodes	262	322	287
Rule Nodes	20	22	17
RMSE	51.03	52.3	52.12
Resubstitution Error (Percentage)	53.25	51.70	54.05

Between the three types of data:

1. Original data was found to perform the best for created nodes and RMSE.
2. CW-ranked data was found to perform the best for rule nodes and resubstitution error.
3. GA-ranked data was found to perform the best for elapsed training and testing time.

Table 5. Classification experiments using holdout validation method

Measurements	Original Data	CW-Ranked Data	GA-Ranked Data
Elapsed Training Time (Seconds)	367.12	480.8	477.54
Elapsed Testing Time (Seconds)	1.062	1.08	1.09
Created Nodes	291	308	350
Pruned Nodes	273	289	330
Rule Nodes	18	19	20
RMSE	48.702	51.52	57.19
Error Rate (Percentage)	73.60	72.67	62.53

Between the three types of data:

1. Original data was found to perform the best for elapsed training and testing time, created nodes and RMSE.
2. CW-ranked data was found to not be able to perform the best for any of the measurements.
3. GA-ranked data was found to perform the best for rule nodes and error rate.

For overall conclusion in this Group 2 experiments, GA-ranked data was found to perform slightly better as an ISA-filtered data compared to CW-ranked data. Therefore, GA-ranked data is the only ISA-filtered data that will be used in the final groups of experiments together with the original data of a new dataset called steel.

3.3. Group 3

There are two validation methods applied in this group of experiments, they are memory recall and holdout. The classifier is EFuNNs and the dataset is steel dataset. The ISA-filtered data is GA-ranked data.Same with Group 2 experiments, only created nodes and rule nodes will be described.

Table 6. Classification experiments using memory recall validation method

Measurements	Original Data	GA-Ranked Data
Elapsed Training Time (Seconds)	177.21	184.6
Elapsed Testing Time (Seconds)	3.19	3.91
Created Nodes	700	692
Pruned Nodes	654	659
Rule Nodes	46	33
RMSE	2.46	2.39
Resubstitution Error (Percentage)	45.80	48.17

Between the two types of data:

1. Original data was found to perform the best for elapsed training and testing time, rules nodes and resubstitution error.
2. GA-ranked data was found to perform the best for created nodes and RMSE.

Table 7. Classification experiments using holdout validation method

Measurements	Original Data	GA-Ranked Data
Elapsed Training Time (Seconds)	176.5	198.37
Elapsed Testing Time (Seconds)	0.865	1.28
Created Nodes	714	695
Pruned Nodes	673	655
Rule Nodes	41	40
RMSE	2.51	2.35
Error Rate (Percentage)	47.76	46.21

Between the two types of data:

1. Original data was found to perform the best for elapsed training and testing time and rules nodes.
2. GA-ranked data was found to perform the best for created nodes, RMSE and error rate.

For overall conclusion in this Group 3 experiments, original data was found to perform slightly better than GA-ranked data. GA-ranked data was consistent though in maintaining the smallest number of created nodes at the point of initial creation and making EFuNNs as the most fitting model.

4. CONCLUSION

This work aims to evaluate how well FR is when applied in ANNs-based classifiers, especially EFuNNsand whether the efficiency and accuracy are improved. For efficiency, the results can be found at the measurements of elapsed training and testing time, created nodes and rule nodes. As for accuracy, the measurements are RMSE, resubstitution and error rates.

Based on the results described in Group 1, 2 and 3 experiments, when FR is applied, some efficiency and accuracy are noticeable. Therefore, the classifiers are improved given the groups of experiments conducted.For future work, FS will be applied to strengthen the results from this work.

5. ACKNOWLEDGEMENTS

This work was supported by the Research Initiative Grant Scheme (RIGS15-073-0073) from the International Islamic University Malaysia (IIUM).

6. REFERENCES

[1] Olden J D, Joy M K, Death R G. An accurate comparison of methods for quantifying variable importance in artificial neural networks using simulated data. Ecological Modelling, 2004, 178(3):389-397

[2] Garson DG. Interpreting neural network connection weights.AI Expert, 1991, 6(7):47-51

[3] Sung A H. Ranking importance of input parameters of neural networks. Expert Systems with Applications, 1998, 15(3):405-411

[4] Watts M J. A decade of Kasabov's evolving connectionist systems: A review. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 2009, 39(3):253-269

[5] Gevrey M, Dimopoulos I, Lek S. Review and comparison of methods to study the contribution of variables in artificial neural network models. Ecological Modelling, 2003, 160(3):249-264

[6] Guyon I, Elisseeff A. An introduction to variable and feature selection. Journal of Machine Learning Research, 2003, 3:1157-1182

[7] Liu H., Motoda H.Feature selection for knowledge discovery and data mining. Boston: Kluwer Academic Publishers, 1998

[8] Webb A. R., Copsey K. D. Statistical pattern recognition.New Jersey: John Wiley and Sons, 2011

[9] Coatelen J, Albouy-Kissi A, Albouy-Kissi B, Coton JP, Maunier-Sifre L, Joubert-Zakeyh J, Dechelotte P, Abergel A. A subset-search and ranking based feature-selection for histology image classification using global and local quantification. In IEEE International Conference on Image Processing Theory, Tools and Applications, 2015, pp. 313-318

[10]Sharma N, Saroha K. A novel dimensionality reduction method for cancer dataset using PCA and Feature Ranking. In IEEE International Conference on Advances in Computing, Communications and Informatics, 2015, pp. 2261-2264

[11] Degeest A, Verleysen M, Frénay B. Feature ranking in changing environments where new features are introduced. In IEEE International Joint Conference on Neural Networks, 2015, pp. 1-8

[12]Bravi L, Piccialli V, Sciandrone M. An optimization-based method for feature ranking in nonlinear regression problems. IEEE Transactions on Neural Networks and Learning Systems, 2017, 28(4):1005-1010

[13]Zdravevski E, Lameski P, Kulakov A, Jakimovski B, Filiposka S, Trajanov D. Feature ranking based on information gain for large classification problems with MapReduce. In IEEE Trustcom/BigDataSE/ISPA, 2015, pp. 186-191

[14]Hsieh C T, Shih J L, Lee C H, Han C C, Fan K C. 3D model retrieval using multiple features and manifold ranking. In 8th IEEE International Conference on Ubi-Media Computing, 2015, pp. 7-10

[15]Bountris P, Tsirmpas C, Haritou M, Pouliakis A, Kouris I, Karakitsos P, Koutsouris D. An ensemble feature ranking framework for the assessment of the efficacy of cervical cancer detection tests and human papillomavirus genotypes in the detection of high-grade cervical intraepithelial neoplasia and cervical carcinoma. In IEEE 15th International Conference on Bioinformatics and Bioengineering, 2015, pp. 1-5

[16]Yan F, Men A, Yang B, Jiang Z. An improved ranking-based feature enhancement approach for robust speaker recognition. IEEE Access, 2016, 4:5258-5267

[17]Sahu M, Sharma S, Raj V, Nagwani N K, Verma S. Impact of Ranked Ordered Feature List (ROFL) on classification with visual data mining techniques. In IEEE International

Conference on Electrical, Electronics, and Optimization Techniques, 2016, pp. 3183-3188

[18]Haykin S.Neural networks: A comprehensive foundation. New York: MacMillan College Publishing, 1994

[19]Noriega L. Multilayer perceptron tutorial. Staffordshire University, 2005, https://s3.amazonaws.com/academia.edu.documents/32342959/mlp.pdf?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1504361103&Signature=iyMETxTvvkLh8SpMAM%2BEcSVwuIM%3D&response-content-disposition=inline%3B%20filename%3DMultilayer_Perceptron_Tutorial.pdf

[20]Pal S K, Mitra S. Multilayer perceptron, fuzzy sets, and classification. IEEE Transactions on Neural Networks. 1992, 3(5):683-697

[21]Kasabov N.Evolving connectionist systems: The knowledge engineering approach. New York: Springer-Verlag, 2007

[22]Kasabov N K. The ECOS framework and the ECO learning method for evolving connectionist systems. Journal of Advanced Computational Intelligence and Intelligent Informatics, 1998, 2(6):195-202

[23]Cai J, Zheng P, Qaisar M, Luo T. Prediction and quantifying parameter importance in simultaneous anaerobic sulfide and nitrate removal process using artificial neural network. Environmental Science and Pollution Research, 2015, 22(11):8272-8279

[24]Ibrahim O M. Evaluating the effect of salinity on corn grain yield using multilayer perceptron neural network. Global Journal of Advanced Research, 2015, 2:400-411

[25]Khuntia S. Modelling of geotechnical problems using soft computing. Master thesis, Rourkela: National Institute of Technology, 2014

[26]Watts M J, Worner S P. Using artificial neural networks to determine the relative contribution of abiotic factors influencing the establishment of insect pest species. Ecological Informatics, 2008, 3(1):64-74

[27]Olanrewaju O A, Mbohwa C. Evaluating factors responsible for energy consumption: Connection weight approach. In IEEE Electrical Power and Energy Conference, 2016, pp. 1-5

[28]Netsanet S, Zhang J, Zheng D, Hui M. Input parameters selection and accuracy enhancement techniques in PV forecasting using Artificial Neural Network. In IEEE International Conference on Power and Power and Renewable Energy, 2016, pp. 565-569

[29]Chang J, Yan L, Liu Y, Liu Y. Feature variable selection based on variable contributions in artificial neural. In R. Zhu, & Y. Ma (Eds.), Information engineering and applications. London: Springer, 2012, pp. 1483-1489

[30] Weckman G R, Millie D F, Ganduri C, Rangwala M, Young W, Rinder M, Fahnenstiel G L. Knowledge extraction from the neural 'black box'in ecological monitoring. Journal of Industrial and Systems Engineering, 2009, 3(1):38-55

[31] Zhou B, Vogt R D, Lu X, Xu C, Zhu L, Shao X, Liu H, Xing M. Relative importance analysis of a refined multi-parameter phosphorus index employed in a strongly agriculturally influenced watershed. Water, Air, and Soil Pollution, 2015, 226(3):1-13

[32] Mouton A M, Dedecker A P, Lek S, Goethals P L. Selecting variables for habitat suitability of Asellus (Crustacea, Isopoda) by applying input variable contribution methods to artificial neural network models. Environmental Modeling and Assessment. 2010, 15(1):65-79

[33] Lal T N, Chapelle O, Schölkopf B. Combining a filter method with SVMs. In I. Guyon, M. Nikravesh, S. Gunn, &L. A. Zadeh (Eds.), Feature extraction. Berlin: Springer, 2006, pp. 439-445

[34] Kumar L, Rath S K. Application of genetic algorithm as feature selection technique in development of effective fault prediction model. In IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics Engineering, 2016, pp. 432-437

[35] Vergara A, Vembu S, Ayhan T, Ryan M A, Homer M L, Huerta R. Chemical gas sensor drift compensation using classifier ensembles. Sensors and Actuators B: Chemical, 2012, 166:320-329

[36] Rodriguez-Lujan I, Fonollosa J, Vergara A, Homer M, Huerta R. On the calibration of sensor arrays for pattern recognition using the minimal number of experiments. Chemometrics and Intelligent Laboratory Systems, 2014, 130:123-134

[37] Research Center of Sciences of Communication. Home.Semeion: Rome, 2010

[38] Bache K, Lichman M. UCI machine learning repository. Oakland: University of California, 2013

[39] Nguyen D, Widrow B. Improving the learning speed of 2-layer neural neural network by choosing initial values of the adaptive weights. Proceedings of IEEE International Joint Conference on Neural Networks, 1992, 5(4):595-603

[40] Beale M H, Hagan M T, Demuth H B. Neural network toolbox™ user's guide. Massachusetts: Mathworks Inc., 2015

[41]Pavelka A, Procházka A. Algorithms for initialization of neural network weights. In MATLAB 12th Annual Conference, 2004, pp. 453-459

[42] Deplano I, Squillero G, Tonda A. Anatomy of a portfolio optimizer under a limited budget constraint. Evolutionary Intelligence, 2016, 9(4):125-136

[43] Aulova A, Govekar E, Emri I. Determination of relaxation modulus of time-dependent materials using neural networks. Mechanics of Time-Dependent Materials, 2017, 21(3):331-349

[44] Zhang J, Lv Y, Chang S, Wang H, He J, Huang Q. Prior knowledge input neural network method for GFET description. Journal of Computational Electronics, 2016, 15(3):911-918

[45] Javed K, Gouriveau R, Li X, Zerhouni N. Tool wear monitoring and prognostics challenges: a comparison of connectionist methods toward an adaptive ensemble model. Journal of Intelligent Manufacturing, 2016:1-8

[46] Olden J D, Jackson D A. Illuminating the "black box": A randomization approach for understanding variable contributions in artificial neural networks. Ecological Modelling, 2002, 154(1):135-150

[47] Yassin I M, Jailani R, Ali M S, Baharom R, Hassan A H, Rizman Z I. Comparison between cascade forward and multi-layer perceptron neural networks for NARX functional electrical stimulation (FES)-based muscle model. International Journal on Advanced Science, Engineering and Information Technology, 2017, 7(1):215-221

How to cite this article:
Hassan R, Al-Shaikhli I F T, Ahmad S. Input significance analysis: feature ranking through synaptic weights manipulation for anns-based classifiers. J. Fundam. Appl. Sci., 2017, *9(4S), 639-662.*