

## INPUT SIGNIFICANCE ANALYSIS: FEATURE SELECTION THROUGH SYNAPTIC WEIGHTS MANIPULATION FOR EFUNNS CLASSIFIER

R. Hassan<sup>1,\*</sup>, I. F. T. Al-Shaikhli<sup>1</sup> and S. Ahmad<sup>2</sup>

<sup>1</sup>Department of Computer Science, Kulliyah of Information and Communication Technology, International Islamic University Malaysia, 50728 Kuala Lumpur, Malaysia

<sup>2</sup>Department of Mechatronics Engineering, Kulliyah of Engineering, International Islamic University Malaysia, 50728 Kuala Lumpur, Malaysia

Published online: 05 October 2017

---

### ABSTRACT

This work is interested in ISA methods that can manipulate synaptic weights namely Connection Weights (CW) and Garson's Algorithm (GA) and the classifier selected is Evolving Fuzzy Neural Networks (EFuNNs). Firstly, it test FS method on a dataset selected from the UCI Machine Learning Repository and executed in an online environment, record the results and compared with the results that used original and ranked data from the previous work. This is to identify whether FS can contribute to improved results and which of the ISA methods mentioned above that work well with FS, i.e. give the best results. Secondly, to attest the FS results by using a differently selected dataset taken from the same source and in the same environment. The results are promising when FS is applied, some efficiency and accuracy are noticeable compared to the original and ranked data.

**Keywords:** feature selection; feature ranking; input significance analysis; evolving connectionist systems; evolving fuzzy neural network; connection weights; Garson's algorithm.

---

Author Correspondence, e-mail: [hrai@iium.edu.my](mailto:hrai@iium.edu.my)

doi: <http://dx.doi.org/10.4314/jfas.v9i4s.39>

---

## 1. INTRODUCTION

Today's technological, social and economic lifestyle and consumer preferences that are dealing with data are changing rapidly and already moving towards the big data phenomenon. Big data refers to datasets of both structured and unstructured types, which are so large or complex and caused the traditional data processing algorithms or software to be inadequate when used to process them. The following are some of the factors that contribute to big data [1]:

1. There exist embedded sensors at everywhere, from smart houses to automobiles. These sensors are producing continuous streams of data in real-time.
2. There were over 5 billion mobile devices in use in 2010. The mobile users are 70% more active on Facebook and Twitter than non-mobile users.
3. The price of storage is falling while capacity is increasing. The data centers are being built all over the world to help companies monetize their overflowing data.

The speed of which the amount of data generated every day is fast [2-5]. There are three characteristics that defined big data: volume, variety and velocity [6]. Volume refers to the arduous amount of data that is being created is actually doubled than the amount of the available storage and 95% of this data is complex or unstructured in nature[1]. Variety refers to the number of types of data and velocity refers to the speed of data processing.

The general challenge with big data is, when volume and variety of the data increase which in return can cause the data velocity to be decreased, the current data processing techniques and technologies may not be able to handle storage and processing of the data [2].

Therefore, there must be a way to overcome this problem. As this work is interested in a specific data processing task, i.e. classification, it is important for classifiers to incorporate method(s) on understanding the role of each of the input data/feature in the dataset before being processed by the system. By having this understanding, it can help to identify redundant and irrelevant data and hence, among others, can speed up the processing time.

This research interest is called Input Significance Analysis (ISA). The definitions of ISA may be as follow:

1. The methods of establishing the significance or rank of each of the input data variable of a Multi-Layer Perceptron (MLP) [7, 51].

2. The procedures for estimating the relation contribution of each of input data variable [8]. Additionally, the process of identifying best data may also be called variable or feature selection (FS). FS refers to the feature irrelevance in which when removed, the remaining features will improve, be the measure of accuracy, consistency, information, distance or dependence [9]. Another simple definition of FS is to reduce the number of variables or input features or input neurons[10].

In conclusion, the ability to identify the best data that is relevant to the problem in hand may directly contribute to faster training (by reducing the network processing time), reducing the size of the network and possibly producing more accurate results [11].

There is a process prior to FS. It is called feature ranking (FR). According to [9], FR is referring to a ranked list of features ordered according to evaluation measures. In this research, the evaluation measures are the ISA techniques selected. FS is then applied on the ranked datasets produced by FR.

The ISA techniques selected are Connection Weights (CW) and Garson's Algorithm (GA). They are specifically chosen because they allow the synaptic weights to be manipulated to create the FR data. The synaptic weights can be obtained from the two ANNs-based classifiers selected; Multi-Layer Perceptron (MLP) and Evolving Fuzzy Neural Networks (EFuNNs).

In the previous work entitled "Input Significance Analysis: Feature Ranking through Synaptic Weights for ANNs-Based Classifiers", it has been established that between the two ANNs-based classifiers, EFuNNs was found to works best with FR data.

There are two motivations for this work. The first motivation is the relationship between big data and classifiers. For data classification purpose, the big dataset can sometimes be hard to deal with in the sense that it requires the classifiers to produce higher accuracy percentage, despite whether the dataset itself have redundant or irrelevant information in the first place. The massive size will cause the classifiers to process longer than necessary. Furthermore, the redundant or irrelevant data may misguide the learning classification algorithms to learn the random error or noise related to them. In some scenarios, this may also cause the data to be overfitted [9]. All of this added complexity to the dataset.

A similar recent work inspired by the same motivation is about making trillion correlations feasible in feature grouping and selection. This work exploited the mechanisms of feature

correlations via feature groups that capture new salient characteristics of the data, especially in big dimensionality which involves general classification tasks. The proposed approach is a general feature grouping and selection framework, which considers an explicit incorporation of correlation measures as constraints in the learning model for different types of learning problem. An efficient embedded feature selection strategy is then designed to remove large numbers of non-contributing correlations that could confuse the classifier while identifying the informative feature groups. The results verified the high efficacies of the proposed methodology, which makes trillion correlations feasible when dealing with big dimensional data [12].

Other than that, there is a much recent work that highlighted the challenges of FS for big data analytics. This work can be related to the previous FR work mentioned earlier. Below are the relevant challenges [13]:

1. Structured features in which most of the existing FS algorithms are designed for generic data and based on the strong assumption that features do not correlate, the FR method applied can be used to tackle this challenge because FR can provide meanings or importance of each of the input features.
2. Streaming data and features, in which when data that needs to be processed in real time to gain insights and the data size and features are unknown by applying FR, some mechanisms of control over the data size and a number of features can be enforced.
3. Scalability, in which the dataset can be increased in tremendous growth, FR also can be used to provide a control mechanism to the initial number of features to be processed.

The second motivation is the relationship between big data and online environment. Nowadays, with almost all information can be found and uploaded into Internet, clouds and online data storage, the dataset can grow larger in sync with the online processes and this can pose even greater problems for classifiers. In short, it means the already massive dataset will be growing in very short time. For examples [1]:

1. Weather conditions (temperature, wind speed, humidity, pressure) taken every second from every weather monitoring station on the globe.
2. DNA sequencing of human and other genomes.

3. Consumer data (what products are sold, at what price and which location) from every store in the world.
4. Energy consumption each minute, from every meter, from every building and home in the world.

Other than that, the online processes also demand accurate real-time results. Therefore, some filtering must be done on the dataset to ensure the only valid information is processed in a particular classification task.

Going back to the first motivation, it is worth to highlight and describe feature extraction (FE).

FE and FS have the same purpose, which is to reduce the number of variables or input features or input neurons [10]. Some of the recent works of FS are:

1. The use of FS for the multiclass datasets, whereby the dataset with the selected feature is tested for classification accuracy with cross-validation approach [14].
2. A FS algorithm called multi-objective genetic local search (MOGLS) that integrates a 3-objective genetic algorithm with a local search heuristic to find feature subsets with the maximum prediction accuracy, the smallest sizes and the minimum redundancy [15].
3. An anomaly detection IDS using Genetic Algorithm (GA) and Support Vector Machine (SVM) with FS that is based on genetic and a fitness function that reduce the dimension of the data, increase true positive detection and simultaneously decrease false positive detection [16].
4. The application of FS as a technique that removes noisy and redundant features to improve the accuracy and generalizability of a prediction model [17].
5. In EEG signals classification, whereby the proposed FS method is based on elastic net and Least Absolute Shrinkage and Selector Operator (LASSO) to optimally select between spatial and temporal features [18].
6. The use of FS with Deep Belief Networks (DBN) in sentiment classification where the FS is a chi-squared based selection that reduced the complexity of the vocabulary input (by filtering some of the irrelevant features) and resulted in the learning phase of DBN to be more efficient [19].

7. The novel FS algorithm that is based on high dimensional model representation, which is effective in modeling input-output system behaviors [20].
8. The use of FS in Brain-Computer Interface (BCI) in which six FS methods (all separately applied to EEG data):CFS, Consistency,ReliefF, mRmR, C4.5 and GA are evaluated andcompared for motor imagery task classification [21].
9. FS applied in text classification, whereby the authors proposed a general importance weighted FS strategy and tested on three text classification datasets [22]. Since it is on importance value, the weighted FS can also be considered as FR in the previous work.
10. The authors proposed an online weighted ensemble of one-class SVMs that is able to select suitable features for Background Subtraction (BS). Due to the online nature, it is also able to update the relative importance of each feature over time [23].
11. A specific FS that can control redundancy by using fuzzy rule based framework. It works by imposing a penalty on the use of dependent/correlated features during system identification along with FS. It can choose good features, discard indifferent and derogatory features, and can control the level of redundancy in the set of selected features [24].
12. FS by using GA is applied to detect botnets at the phase of Command and Control (C&C), i.e. set of features that can give the highest detection rate [25].
13. The use of FS in Convolutional Neural Networks (CNNs) to reduce the dimensions of features in CNNs [26].
14. Three FS algorithms as a FS-based clustering are used to identify the determinant factor of breast cancer in Indonesia. The results showed that only 14 out of the original 21 attributes have the highest contribution factor of breast cancer [27].

Technically, the dataset that has been reduced is called dataset in a reduced number of dimensions. With this, there exist techniques called ordination methods or geometrical methods that represent datasets in a reduced dimension such as principal component analysis and multidimensional scaling [10].

However, this work is going for a different approach. Before a dataset is being reduced by FS, FR will be applied to it. As part of ISA implementation, FR will rank the input features of the dataset by importance value for the purpose of identifying which features are important or useful before deciding to select the ones that will be removed from the dataset.

FE is also strongly related to knowledge discovery or KDD. KDD is referring to the nontrivial process of identifying valid, novel, potentially useful and ultimately understandable patterns in data [28]. The KDD process can be decomposed into the following steps [28]:

1. Data selection: The purpose is to select or create a target dataset. This target dataset may come from the merger of several datasets.
2. Data cleaning or pre-processing: The purpose is to reduce or remove any noises and missing values.
3. Data reduction: The purpose is to create a dataset that only contains useful features. This also means the irrelevant features are removed.
4. Data mining: It refers to the extraction of hidden predictive information from large databases. Here, the choice of algorithm depends on many factors. Among others, the source of the dataset and the values it contains.
5. Interpretation or evaluation: Validation, usefulness, novelty and simplicity are the measurements that will be used to evaluate the discovered knowledge.

Ideally, any classifiers should be able to process any datasets. However, to represent the dataset in a more stable representation and possibly improve classification performance, FS is needed [29-30]. There are three categories of FS approaches. They are [10]:

1. Filter methods: The filtering out of the poor information variables is applied before the classification task is executed. This method is independent of classifier learning.
2. Wrapper methods: This method evaluates the variables within the classification algorithm. This means that this method is dependent on the classifier.
3. Embedded methods: The feature selection technique selected is built inside the classifier. This means that this method is also dependent on the classifier. But from above, the search to find the optimal dataset can be seen as the combination of the features selection and classifier models.

This work on the surface seems to apply the filter method. Nevertheless, because of the use of EFuNNs as the classifier, once the filtered/FR dataset is fed into EFuNNs, the learning capability of EFuNNs will enhance the dataset as it is being processed until the end. Therefore, it can be considered as embedded method as well.

Going back to the second motivation, it is worth to highlight and describe the Evolving

Connectionist Systems (ECoSs) and EFuNNs.

ECoSs is a parent concept of EFuNNs. In general, ECoSs can be referred to as a family of constructive ANNs algorithms. Due to its ANNs-based architecture, it means that at its core, an ECoSs has a connectionist structure that has information processing units as neurons and also available connections between them [31].

ECoSs is an example of Evolving Intelligent Systems (EISs). “Evolving” here does not refer to something that is progressing gradually such as through simulated evolution processes found in Genetic Algorithms (GA), rather it refers to something that is changing over time [7]. Its foundation is laid in the seven requirements or abilities of Intelligent Information Systems (IISs) [32-33] (as cited in [7]):

1. Able to learn huge data at high speed.
2. Able to gradually adapt to new data in actual time.
3. Employs open structure where the additional features such as inputs and outputs can easily be added.
4. Able to justifiably track and retrieve the previously seen data.
5. Continual improvement from the beginning to the end of the system’s life.
6. By using rule extraction as an example, the system can analyze and explain itself.
7. Able to characterize a variety of elements of data, such as spatial and temporal elements.

ECoSs has the attributes of adaptive and incremental learning. Its structure and functionality as part of knowledge can also evolve [31]. Not only the structure of an ECoS can change over time gradually, but its set of input variables (features) can behave the same too [34].

EFuNNs is the advanced concept of ANNs that combines ANNs with Fuzzy Logic (FL). 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 [31]. In simple terms, it is characterized by embedded fuzzy logic elements [32].

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 rules aggregation. This technique allows several rule nodes to be merged into one [31]. 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. EFuNNs is chosen as the ECoSs classifier to be used in the first work because it has all the elements listed below [35]:

1. Adaptive learning
2. Incremental learning
3. Lifelong learning
4. Online learning
5. Constructivist structural learning
6. Selectivist structural learning
7. Knowledge-based learning neural network (KBNN)

This simply means EFuNNs has all the capability as an online, knowledge-based intelligent systems [35].

This work aims to combine FS and classifiers learning ability as one computational model. The computational model proposed by this work can be referred to as an online FS based on ISA for ECoSs, which consists of FR and FS. The definition of an online environment here is one that is performing FS and classifiers learning concurrently. This work can take advantage of the readily available online environment of EFuNNs (by setting the mode), to link FS (via ISA methods) with learning classifiers. Thus, a more enhanced classifier can be created. Similar works of ISA methods and techniques in ECoSs are as the following.

**Table 1.** ISA methods and techniques in ECoSs

No.	Techniques	Work Summary
1.	[36] Incremental Recursive Fisher Linear Discriminant (IRFLD)	The authors proposed an improvement of IRFLD for online feature extraction. IRFLD is compared with the conventional Incremental Linear Discriminant Analysis (ILDA) and performed better than ILDA.
2.	[37] Incremental Feature	Weights of [0,1] were assigned to features, according to

- 
- Weighting in Evolving Fuzzy Classifiers their importance level. These weights are permanently updated during the on-line mode.
3. [38] Incremental Recursive Fisher Linear Discriminant (IRFLD) The authors proposed a new online feature extraction by using IRFLD. IRFLD is compared with the conventional Linear Discriminant Analysis (LDA) and found to be performed better than LDA.
4. [39] Incremental Principal Component Analysis (IPCA) The authors discussed the current IPCA and Chunk IPCA (which enhances the learning speed) and, how a classification system is constructed.
5. [40] Growing Gaussian mixture model (GGMM), resource allocating neural network (RAN) and Incremental Feature Selection This paper introduced hybrid evolving architecture (GGMM and RAN) for dealing with incremental learning. Incremental feature selection was used to choose meaningful features.
6. [41] Evolving Clustering Method (ECM) and Incremental Principal Component Analysis (IPCA) The IPCA and ECM are combined to make up a new adaptive ECoSs specified for pattern classification systems.
7. [42] Incremental Linear Discriminant Analysis (ILDA) (Sequential and Chunk) The data streams are classified by using the selected ISA techniques.
8. [43] Incremental Principal Component Analysis (IPCA) and Resource Allocating Network with Long-Term Memory An extended version of IPCA and RAN-LTM are effectively combined to implement the same idea as no. 10.
-

(RAN-LTM)

- [44] Incremental Principal Feature selection and classifier learning that are
9. Component Analysis concurrently carried out on-line is the proposed new (IPCA) and Evolving concept for pattern classification systems in this work.
- Clustering Method (ECM)

## 2. MATERIAL AND METHOD

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

### 2.1. Experimental Research Design

The following Fig. 1 shows the experimental design that includes the previous work (FR) as mentioned in section 1.

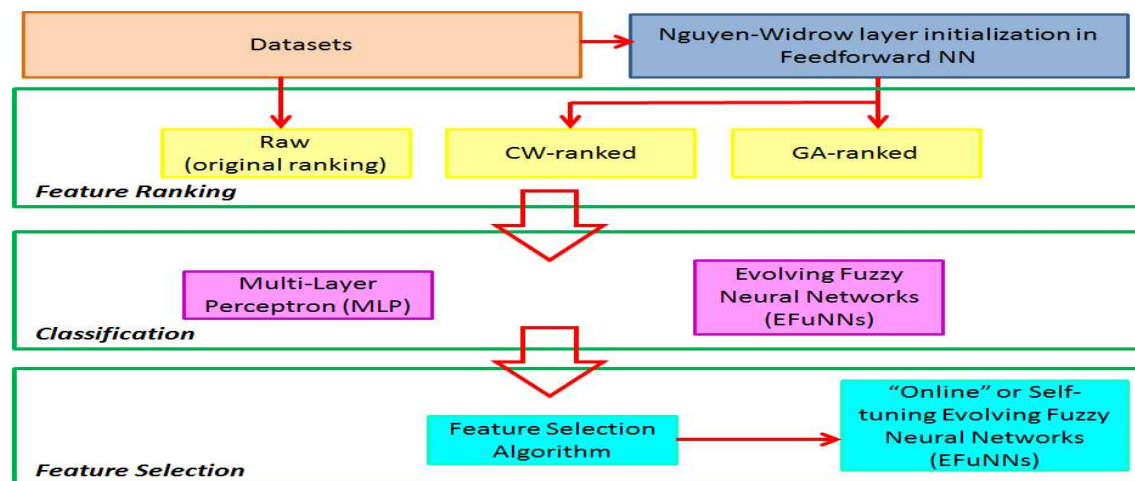


Fig.1. Experimental research design

### 2.2. Working Environment

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

### 2.3. Data

There are two datasets used. The first one is called gas sensor array drift (or shortly as gas sensors) [45-46]. The second one is called steel plate's faults dataset (or shortly as steel) [47] and both are from UCI Machine Learning Repository [48].

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 gasses, at different concentration levels. There are ten batches (ten data collections),

organized into 128 independent variables and 6 dependent variables. The 6 dependent variables are [45-46]:

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 [47]:

1. Pastry
2. Z\_Scratch
3. K\_Scratch
4. Stains
5. Dirtiness
6. Bumps
7. Other\_Faults

#### **2.4. Validation Methods**

There are two validation methods used in this research. They are:

1. Memory recall
2. 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.

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

**2.6. CW**

The formula is referred below [50]. First, the product between the raw input-hidden and hidden-output connection weights between each input neuron/feature and output neuron/feature are calculated [49]. After that, the products across all hidden neurons/features are summed up [49].

$$\text{Input}_x = \sum_{Y=A}^E \text{Hidden}_{XY} \tag{1}$$

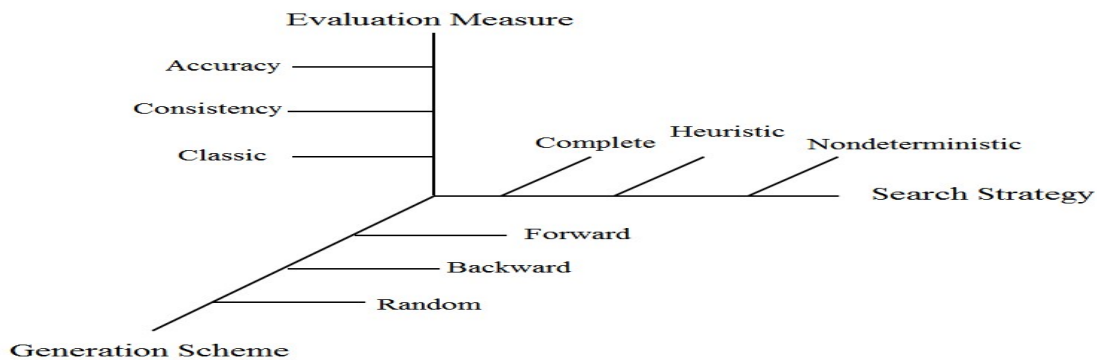
**2.7. GA**

The formula is referred below [50]. First, the partitioning into components is applied to the hidden-output connection weights [49]. After that, each input neuron/feature that uses absolute values of connection weights is then linked to these components [49].

$$\text{Input}_x = \sum_{Y=A}^E \frac{|\text{Hidden}_{XY}|}{\sum_{Z=1}^5 |\text{Hidden}_{ZY}|} \tag{2}$$

**2.8. Performance Measurements**

There are three principal dimensions of FS: search strategy, evaluation measure and feature generation scheme [9]. 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. Random cut-off feature number, cutfeat: This signifies the start number of feature or column to be deleted. Once a feature is marked as the cut-off number, the marked feature itself until the last feature (to the right) will be deleted.
2. Elapsed training time, etrt (seconds): This measurement is used to observe the network processing time during training.
3. Elapsed testing time, ett (seconds): This measurement is used to observe the network processing time during testing.
4. Created nodes, cn: This measurement refers to the number of nodes created initially at the start of the classification process.
5. 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.

$$pn = f(cn) \quad (3)$$

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

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

$$ln = cn - pn \quad (4)$$

7. 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:

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

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

8. Resubstitution Error, 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:

$$re = \text{predicted values}(\text{data}_x) - \text{observed values}(\text{data}_x) \quad (6)$$

where  $\text{data}_x$  = refers to the whole dataset.

9. 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:

$$re = \text{predicted values (data}_x) - \text{observed values (data}_y) \quad (7)$$

where  $\text{data}_x$  = refers to, say 30% of the dataset and  $\text{data}_y$  = refers to the remaining 70% of the dataset

### 2.9. FS Algorithm

The following is the algorithm to decide which feature (as a start feature) will be deleted and the cycle of FS decision-making:

Step 1: Choose a random feature number.

Step 2: Delete the selected feature number up to the end feature in the dataset (to the right).

Step 3: Save as a different data file.

Step 4: Process the new data file in an online or self-tuning EFuNNs.

Step 5: Records the results obtained.

Step 6: Compare the results in step 5 with the previous results. For initial comparison, the results in step 5 will be compared with the results obtained in Group 2 experiments that have been conducted in previous work mentioned in section 1.

1. If any of the measurements has improved results when compared to the previous results:
  - a. Choose a feature number lesser than the feature number chosen in step 1.
  - b. Repeat Steps 1 to 6.
2. General guideline: if all measurements obtained poorer results when compared to the previous results:
  - a. Choose a feature number greater than the feature number chosen in step 1.
  - b. Repeat steps 1 to 6.

Step 7: Stop until satisfied results are obtained, or no more tries or experiments left.

For this work, since the FS decision-making process is a manual process, therefore, only five tries or experiments are allocated. It is hoped that this number of tries or experiments can give some initial insights about the effectiveness of FS in EFuNNs.

Additionally, since this work is about classification task and to ensure that EFuNNs is a better fitting model, the priorities are to get the minimum classification error and low RMSE. Therefore, for step 6 of number 2, if the results for either resubstitution error or error rate and RMSE are poorer than the previous results, then the next feature number selected will be greater than the current one.

## 2.10. Experiments

There are two groups of experiments conducted.

1. Group 1: To test FS method

This group of experiments applied FS to the CW- and GA-ranked datasets in an online environment or self-tuning mode.

2. Group 2: To test FS method by using a different dataset

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

## 3. RESULTS AND DISCUSSION

### 3.1. Results from Previous Work

The following results captured the effect(s) of using the ISA-filtered dataset (from the gas sensors dataset) and in an online environment or self-tuning EFuNNs. They will be used as an initial comparison with the results from the Group 1 experiments in this work.

**Table 2.** Classification experiments using memory recall validation method

Measurements	Original Data	CW-Ranked Data	GA-Ranked Data
Elapsed Training Time (Seconds)	447.37	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



**Table 3.**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)	1.06	1.08	1.1
Created Nodes	291	308	350
Pruned Nodes	273	289	330
Rule Nodes	18	19	20
RMSE	48.70	51.52	57.19
Error Rate (Percentage)	73.60	72.67	62.53

The following results captured the effect(s) of using the GA-ranked data (from the steel dataset) and in an online environment or self-tuning EFuNNs. They will be used as an initial comparison with the results from the Group 2 experiments in this work.

**Table 4.**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	2.38
Resubstitution Error (Percentage)	45.80	48.17

**Table 5.**Classification experiments using holdout validation method

Measurements	Original Data	GA-Ranked Data
Elapsed Training Time (Seconds)	176.5	198.37
Elapsed Testing Time (Seconds)	0.86	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

### 3.2. Group 1

There are two validation methods applied in this group of experiments and they are memory recall and holdout. The dataset used is gas sensors, the ISA-filtered data are CW- and GA-ranked 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 lesser number of nodes, and most probably the training and testing time also will be reduced compared to the first classification process.

**Table 6.** Classification experiments using memory recall validation method for CW-ranked data

Measurements	Ex. 1	Ex. 2	Ex. 3	Ex. 4	Ex. 5
Random Cut-Off Feature No.	111	96	82	85	
Elapsed Training Time (Seconds)	457.63	391.19	360.51	321.09	
Elapsed Testing Time (Seconds)	4.99	3.35	2.97	3.46	
Created Nodes	318	308	326	283	
Pruned Nodes	301	291	306	269	
Rule Nodes	17	17	20	14	
RMSE	55.82	56.61	49.97	50.99	
ResubstitutionError (Percentage)	79.32	57.29	73.45	50.01	

The created node for CW-ranked data was 344 (see Table 2-CW-ranked data). The created node at the first experiment stated above was 318 (see Table 6-Ex. 1). Therefore, this allows for the next experiment to be conducted, with a lesser feature number selected.

At experiment 4, satisfied results are obtained. The improvements can be seen at the training time, RMSE and resubstitution error measurements.

1. The original data results of the three measurements were 447.37, 51.03 and 53.25 respectively (see Table 2-Original Data).

2. The CW-ranked data results of the three measurements were 436.1, 52.3 and 51.70 respectively (see Table 2-CW-Ranked Data).
3. The improved results from experiment 4 were 321.09, 50.99 and 50.01 respectively (see Table 6-Ex. 4).

**Table 7.** Classification experiments using memory recall validation method for GA-ranked data

Measurements	Ex.1	Ex. 2	Ex. 3	Ex. 4	Ex.5
Random Cut-Off Feature No.	82	72	77	67	84
Elapsed Training Time (Seconds)	252.49	223.28	272.17	209.82	264.82
Elapsed Testing Time (Seconds)	2.79	2.68	2.75	2.97	2.85
Created Nodes	308	277	328	312	304
Pruned Nodes	289	260	309	292	283
Rule Nodes	19	17	19	20	21
RMSE	53.23	57.22	55.42	52.99	54.29
Resubstitution Error (Percentage)	66.23	68.38	55.57	78.68	50.78

The training and testing time for GA-ranked data was 442.4 and 2.88 respectively (see Table 2-GA-ranked data). The training and testing time at the first experiment stated above was 252.49 and 2.79 respectively (see Table 8-Ex. 1). Therefore, this allows for the next experiment to be conducted with a lesser feature number selected.

At experiment 5, the number of tries has run out. The final results are quite satisfactory though. The improvements can be seen at the training and testing time and resubstitution error measurements.

1. The original data results of the three measurements were 447.37, 3.13 and 53.25 respectively (see Table 2-original data).
2. The GA-ranked data results of the three measurements were 442.4, 2.88 and 54.05 respectively (see Table 2-GA-ranked data).
3. The improved results from experiment 5 were 264.82, 2.85 and 50.78 respectively (see Table 7-Ex. 5).

The conclusion for the memory recall validation method is based on the results. There is an inclination towards CW-ranked data that it can work well with this validation method.

However, the classification error from the previous work highlighted that for CW-ranked data, the result is improved from 51.70 (see Table 2-CW-ranked data) to 50.01 (see Table 6-Ex. 4). For GA-ranked data, the classification error is improved from 54.05 (see Table 2-GA-ranked data) to 50.78 (see Table 7-Ex. 5) which indicates a bigger improvement compared to CW-ranked data.

Moreover, the number of deleted features for GA-ranked data is 45 (equals to 35.16%, see Table 8-Ex. 5) and for CW-ranked data is 44 (equals to 34.38%, see Table 6-Ex. 4).

Therefore, at this stage, both CW-ranked and GA-ranked data can be considered as suitable to be processed through memory recall validation method.

**Table 8.** Classification experiments using holdout validation method for CW-ranked data

Measurements	Ex. 1	Ex. 2	Ex. 3	Ex. 4	Ex. 5
Random Cut-Off Feature No.	102	97			
Elapsed Training Time (Seconds)	274.99	263.47			
Elapsed Testing Time (Seconds)	256	0.88			
Created Nodes	291	312			
Pruned Nodes	273	292			
Rule Nodes	18	20			
RMSE	46.27	46.91			
Error Rate (Percentage)	70.13	52.80			

The training time, created nodes, RMSE and error rate for CW-ranked data was 480.8, 308, 51.52 and 72.67 respectively (see Table 3-CW-ranked data). The training time, created nodes, RMSE and error rate at the first experiment above was 274.99, 291, 46.27 and 70.13 respectively (see Table 8-Ex. 1). Therefore, this allows for the next experiment to be conducted, with a lesser feature number selected.

At experiment 1, satisfied results are obtained. The improvements can be seen at the training and testing time, and RMSE measurements.

1. The original data results of the three measurements were 367.12, 1.06 and 48.70 respectively (see Table 3-original data).
2. The CW-ranked data results of the three measurements were 480.8, 1.08 and 51.52 respectively (see Table 3-CW-ranked data).
3. The improved results from experiment 2 were 263.47, 0.88 and 46.91 respectively (see Table 8-Ex. 2).

**Table 9.** Classification experiments using holdout validation method for GA-ranked data

Measurements	Ex. 1	Ex. 2	Ex. 3	Ex. 4	Ex. 5
Random Cut-Off Feature No.	102	97	92	87	83
Elapsed Training Time (Seconds)	444.89	403.95	338.66	334.09	485.57
Elapsed Testing Time (Seconds)	1.05	0.95	0.92	0.85	0.89
Created Nodes	309	316	294	289	315
Pruned Nodes	291	301	274	274	297
Rule Nodes	18	15	20	15	18
RMSE	54.16	66.40	57.25	60.65	68.62
Error Rate (Percentage)	70.93	69.53	67.60	68.40	55.60

The training and testing time, created nodes and error rate for GA-ranked data was 477.54, 1.1, 350 and 62.53 respectively (see Table 3-GA-ranked data). The training and testing time, created nodes and error rate at the first experiment stated above as 444.89, 1.05, 309 and 54.16 respectively (see Table 9-Ex. 1). Therefore, this allows for the next experiment to be conducted with a lesser feature number selected.

At experiment 5, the number of tries has run out. The final results are quite satisfactory though. The improvements can be seen in the testing time and error rate measurements.

1. The original data results were 1.06 and 73.60 respectively (see Table 3-original data).
2. The GA-ranked data results were 1.1 and 62.53 respectively (see Table 3-GA-ranked data).
3. The improved results from experiment 5 were 0.89 and 55.60 respectively (see Table 9-Ex. 5).

The conclusion for this holdout validation method is based on the results. There is an inclination towards CW-ranked data that it can work well with this validation method.

However, the number of deleted features for GA-ranked data is 46 (equals to 35.94%, see Table 9-Ex. 5) and for CW-ranked data is only 32 (equals to 25%, see Table 8-Ex. 2).

To recall, FS is referring to the number of selected features deleted from the whole ranked dataset. The idea is to create a smaller version of the dataset but with improved results.

In conclusion, for both validation methods, it is concluded that GA-ranked data has performed consistently compared to CW-ranked data.

### 3.3. Group 2

There are two validation methods applied in this group of experiments and they are memory recall and holdout. The classifier selected is an online environment or self-tuning EFuNNs, and the GA-ranked steel dataset is used.

For the memory recall validation method, the percentage of deleted features from Group 1 experiments is 35.16% which is roughly equivalent to selected cut-off feature number 18 (equals to 37.05%) for the steel dataset.

For the holdout validation method, the percentage of deleted features from Group 1 experiments is 35.94%, which is also roughly equivalent to selected cut-off feature number 18 (equals to 37.05%) for the steel dataset.

**Table 10.** Classification experiments using memory recall validation method

Measurements	GA-Ranked Data
Selected Cut-Off Feature No.	18
Elapsed Training Time (Seconds)	106.35
Elapsed Testing Time (Seconds)	4.19
Created Nodes	640
Pruned Nodes	598
Rule Nodes	42
RMSE	2.36
Resubstitution Error (Percentage)	46.57

The results of comparisons between original data (see Table 4-original data) and (new) GA-ranked data (see Table 10-GA-ranked data) are:

1. The results for original data for the training time and RMSE were 177.21 and 2.46 respectively.
2. The improved results for GA-ranked data, for training time and RMSE from the above table were 106.35 and 2.36.

The results of comparisons between (old) GA-ranked data (see Table 4-GA-Ranked Data) and (new) GA-ranked data (see Table 10-GA-ranked data) are:

1. The results for training time, created nodes, RMSE and resubstitution error for GA-ranked data was 184.6, 692, 2.38 and 48.17 respectively.

- The improved results for GA-ranked data for training time, created nodes, RMSE and resubstitution error were 106.35, 640, 2.36 and 46.57 respectively.

**Table 11.** Classification experiments using holdout validation method

Measurements	GA-Ranked Data
Selected Cut-Off Feature No.	18
Elapsed Training Time (Seconds)	121.32
Elapsed Testing Time (Seconds)	0.85
Created Nodes	646
Pruned Nodes	606
Rule Nodes	40
RMSE	2.27
Error Rate (Percentage)	44.98

The results of comparisons between original data (see Table 5-original data) and (old) GA-ranked data (see Table 5-GA-ranked data) and (new) GA-ranked data (see Table 11-GA-ranked data) are:

- The original data results for the training and testing time, created node, RMSE and error rate were 176.5, 0.86, 714, 2.51 and 47.76 respectively.
- The GA-ranked data results for the training and testing time, created node, RMSE and error rate were 198.37, 1.28, 695, 2.35 and 46.21 respectively.
- The improved results for GA-ranked data for the training and testing time, created nodes, RMSE and error rate were 121.32, 0.85, 646, 2.27 and 44.98 respectively.

In this group of experiments, the results showed that when FS is applied, the results are improved as well as consistent due to the same measurements improved from the original and (old) GA-ranked data to the (new) GA-ranked data. For the overall conclusion, the findings are:

- The percentage of deleted features from gas sensor dataset seemed to be applicable to be applied to the steel dataset.
- The results at Table XI showed that FS has successfully improved both original and GA-ranked data results.

3. It is also indicated that the selected features of GA-ranked data are suitable to be applied with holdout validation method.

#### **4. CONCLUSION**

This work aims to evaluate how well FS is when applied to the original and filtered/ranked dataset in an EFuNNs classifier that can be set in an online environment or self-tuning mode, and also 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, the resubstitution and error rates.

Based on the results described in Group 1 and 2 experiments, when FS is applied, some efficiency and accuracy are noticeable. It is also found that GA was the best ISA method compared to CW.

For future work, there are a number of ideas or suggestions on how to improve and expand this research work. A few of the immediate suggestions are as the following:

1. To automate the FS decision-making algorithm with prioritized measurements.
2. To explore other validation methods.
3. To explore other ISA methods.

#### **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] Brenner M. What is big data? Digitalist Magazine. 2012, <http://www.digitalistmag.com/technologies/big-data/2012/05/09/big-data-what-is-it-05326>
- [2] Suthaharan S. Big data classification: Problems and challenges in network intrusion prediction with machine learning. ACM SIGMETRICS Performance Evaluation Review, 2014, 41(4):70-73
- [3] Jadhav D K. Big data: the new challenges in data mining. International Journal of



- 
- Innovative Research in Computer Science and Technology, 2013, 1(2):39-42
- [4] Zaslavsky A, Perera C, Georgakopoulos D. Sensing as a service and big data. In International Conference on Advances in Cloud Computing, 2012, pp. 21-29
- [5] Sagioglu S, Sinanc D. Big data: A review. In IEEE International Conference on Collaboration Technologies and Systems, 2013, pp. 42-47
- [6] Zikopoulos P, Eaton C. Understanding big data: Analytics for enterprise class hadoop and streaming data. New York: McGraw-Hill/Osborne Media, 2011
- [7] 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
- [8] 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
- [9] Liu H., Motoda H. Feature selection for knowledge discovery and data mining. Berlin: Springer Science and Business Media, 1998
- [10] Webb A. R., Copsey K. D. Statistical pattern recognition. New Jersey: John Wiley and Sons, 2012
- [11] Sung AH. Ranking importance of input parameters of neural networks. Expert Systems with Applications, 1998, 15(3):405-411
- [12] Zhai Y, Ong YS, Tsang IW. Making trillion correlations feasible in feature grouping and selection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016, 38(12):2472-2486
- [13] Li J, Liu H. Challenges of feature selection for big data analytics. IEEE Intelligent Systems, 2017, 32(2):9-15
- [14] Dubey VK, Saxena AK, Shrivastava MM. A cluster-filter feature selection approach. In IEEE International Conference on ICT in Business Industry and Government, 2016, pp. 1-5
- [15] Tian D. A multi-objective genetic local search algorithm for optimal feature subset selection. In IEEE International Conference on Computational Science and Computational Intelligence, 2016, pp. 1089-1094
- [16] Gharaee H, Hosseinvand H. A new feature selection IDS based on genetic algorithm and

- SVM. In 8th IEEE International Symposium on Telecommunications, 2016, pp. 139-144
- [17] Osman H, Ghafari M, Nierstrasz O. Automatic feature selection by regularization to improve bug prediction accuracy. In IEEE Workshop on Machine Learning Techniques for Software Quality Evaluation, 2017, pp. 27-32
- [18] Noei S, Ashtari P, Jahed M, Vahdat BV. Classification of EEG signals using the Spatio-temporal feature selection via the elastic net. In 23rd IEEE Iranian Conference on Biomedical Engineering and 2016 1st International Iranian Conference on Biomedical Engineering, 2016, pp. 232-236
- [19] Ruangkanokmas P, Achalakul T, Akkarajitsakul K. Deep belief networks with feature selection for sentiment classification. In 7th IEEE International Conference on Intelligent Systems, Modelling and Simulation, 2016, pp. 9-14
- [20] Taşkın G, Kaya H, Bruzzone L. Feature selection based on high dimensional model representation for hyperspectral images. IEEE Transactions on Image Processing, 2017, 26(6):2918-2928
- [21] Ramos AC, Hernández R G, Vellasco M. Feature selection methods applied to motor imagery task classification. In IEEE Latin American Conference on Computational Intelligence, 2016, pp. 1-6
- [22] Li B. Importance weighted feature selection strategy for text classification. In IEEE International Conference on Asian Language Processing, 2016, pp. 344-347
- [23] Silva C, Bouwmans T, Frélicot C. Online weighted one-class ensemble for feature selection in background/foreground separation. In 23rd IEEE International Conference on Pattern Recognition, 2016, pp. 2216-2221
- [24] Chung IF, Chen YC, Pal N. Feature selection with controlled redundancy in a fuzzy rule based framework. IEEE Transactions on Fuzzy Systems, 2017, 1-15
- [25] Alejandro FV, Cortés NC, Anaya EA. Feature selection to detect botnets using machine learning algorithms. In IEEE International Conference on Electronics, Communications and Computers, 2017, pp. 1-7
- [26] Qian R, Yue Y, Coenen F, Zhang B. Visual attribute classification using feature selection and convolutional neural network. In IEEE 13th International Conference on Signal Processing, 2016, pp. 649-653

- 
- [27] Fahrudin TM, Syarif I, Barakbah AR. The determinant factor of breast cancer on medical oncology using feature selection based clustering. In IEEE International Conference on Knowledge Creation and Intelligent Computing, 2016, pp. 232-239
- [28] Jensen R., Shen Q. Computational intelligence and feature selection: rough and fuzzy approaches. New Jersey: John Wiley and Sons, 2008
- [29] Hoi SC, Wang J, Zhao P, Jin R. Online feature selection for mining big data. In 1st ACM International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications, 2012, pp. 93-100
- [30] Tan M, Tsang IW, Wang L. Towards ultrahigh dimensional feature selection for big data. *Journal of Machine Learning Research*, 15:1371-1429
- [31] Kasabov N. *Evolving connectionist systems: The knowledge engineering approach*. New York: Springer-Verlag, 2007
- [32] Kasabov NK. 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
- [33] Kasabov N. *Evolving connectionist systems: Methods and applications in bioinformatics, brain study and intelligent machines*. New York: Springer-Verlag, 2003
- [34] Kasabov N. Brain-, gene-, and quantum inspired computational intelligence: Challenges and opportunities. In W. Duch, & J. Mańdziuk (Eds.), *Challenges for computational intelligence*. New York: Springer-Verlag, 2007, pp. 193-219
- [35] Kasabov N. Evolving fuzzy neural networks for supervised/unsupervised online knowledge-based learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 2001, 31(6):902-918
- [36] Ohta R, Ozawa S. An improvement of incremental recursive Fisher linear discriminant for online feature extraction. *Electronics and Communications in Japan*, 2013, 96(4):29-40
- [37] Lughofer E. On-line incremental feature weighting in evolving fuzzy classifiers. *Fuzzy Sets and Systems*, 2011, 163(1):1-23
- [38] Ozawa S, Ohta R. Incremental recursive fisher linear discriminant for online feature extraction. In *IEEE Workshop on Evolving and Adaptive Intelligent Systems*, 2011, pp. 70-76
- [39] Ozawa S, Pang S, Kasabov N. Online feature extraction for evolving intelligent systems.

---

In P. Angelov, D. P. Filev, & N. Kasabov (Eds.), *Evolving intelligent systems: Methodology and applications*. New Jersey: John Wiley and Sons, 2010, pp. 151-172

[40] Bouchachia A. An evolving classification cascade with self-learning. *Evolving Systems*, 2010, 1(3):143-160

[41] Ozawa S, Pang S, Kasabov N. On-line feature selection for adaptive evolving connectionist systems. *International Journal of Innovative Computing, Information and Control*, 2006, 2(1):181-192

[42] Pang S, Ozawa S, Kasabov N. Incremental linear discriminant analysis for classification of data streams. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 2005, 35(5):905-914

[43] Ozawa S, Toh SL, Abe S, Pang S, Kasabov N. Incremental learning of feature space and classifier for face recognition. *Neural Networks*, 2005, 18(5):575-584

[44] Ozawa S, Pang S, Kasabov N. A modified incremental principal component analysis for on-line learning of feature space and classifier. In *8th Pacific Rim International Conference on Artificial Intelligence*, 2004, pp. 231-240

[45] Vergara A, Vembu S, Ayhan T, Ryan MA, Homer ML, Huerta R. Chemical gas sensor drift compensation using classifier ensembles. *Sensors and Actuators B: Chemical*, 2012, 166:320-319

[46] 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

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

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

[49] Olden JD, Jackson DA. Illuminating the “black box”: A randomization approach for understanding variable contributions in artificial neural networks. *Ecological Modelling*, 2002, 154(1):135-150

[50] Olden JD, Joy MK, Death RG. An accurate comparison of methods for quantifying variable importance in artificial neural networks using simulated data. *Ecological Modelling*,

2004, 178(3):389-397

[51] [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 selection through synaptic weights manipulation for efunns classifier. *J. Fundam. Appl. Sci.*, 2017, *9(4S)*, 681-709.