MFES FRAMEWORK FOR EFFICIENT FEATURE SELECTION AMONG SUBSYSTEMS IN INTELLIGENT BUILDING

[*]Abba Babakura[1], Abubakar Roko[2], Aminu Bui[3], Ibrahim Saidu[4], Mahmud Ahmad Yusuf[5]

[1,2,3]Department of Computer Science, UDUS

[4]Department of Information and Communication Technology, UDUS

[5]Department of Computer Science, Bayero University Kano

***abba_babakura@yahoo.com**

Abstract. The increasing trend of problem representation and high-dimensional data collection calls for the utilization of feature selection in many machines learning tasks and big data representations. However, identifying meaningful features from thousands of related features in the smart home data which are dissimilar in nature remains a nontrivial task. This has prompted for the deployment of a feature selection algorithm (FSA) that provides two possible solutions. First, to provides an efficient scheme that best optimizes the features for subsystem decisions and second, tackles feature subset selection bias problem. In this paper, a MFES framework for feature selection is proposed that uses a hybrid mechanism to tackle the problem of feature subset selection bias in intelligent building data. The mechanism uses the effectiveness of filters and accuracy of wrappers to obtain significant features for prediction. The proposed MFES framework resulted in 92.17% of accuracy as compared to the baseline approach resulting in 87.21% of accuracy. The experimental results show that efficient and better prediction accuracy can be achieved with a smaller feature set.

Keywords: Feature selection, Machine learning, Simulated annealing algorithm (SAA), F-score, Info-gain.

1 Introduction

Feature selection has been a widely utilized technique for the dimensionality reduction, becoming the main point of research area in Big data, data mining as well as machine learning (Guyon and Elisseeff 2003, Liu and Yu 2005). In machine learning, as data dimensionality increases, the necessary amount of data which provides a dependable analysis also grows exponentially. This phenomenon is termed curse of dimensionality in the problem domain of dynamic optimization by Bellman (Bellman and Dreyfus 2015). A common method for problems related to high dimensional datasets is to forecast the data to smaller number of features that can preserve information as much as possible. Also, there is an increasing difficulty in justifying the results statistically due to sparsity of meaningful data, because of significant growth in dataset dimensionality.

In intelligent building, besides the different communication protocol deployed, the data types utilized are also dissimilar among subsystems. Eventually, such condition would transform the IB into a data-intensive entity as it engrosses exchange of many data types to carry out appropriate interoperation among subsystems. Figure 1 shows the classification of data type's hierarchy for common subsystems interoperation in intelligent building. Based on the hierarchy shown in figure 1, many of these data types are disparate and fixed with respect to subsystems protocol and utilized based on their configured application domains.

By default, each subsystem is associated with their respective data types. Such structured data types are powerful and at same time are most complicated.

The aim of feature selection is finding most relevant features in a problem domain. It significantly improves computational speed and accuracy of prediction. However, identifying meaningful features from thousands of related features in intelligent building dataset which are dissimilar in nature remains a nontrivial task. This has prompted for the deployment of a feature selection algorithm (FSA) that provides two possible solutions. First, to provides an efficient scheme that best optimizes the features for subsystem decisions and second, tackles feature subset selection bias problem.

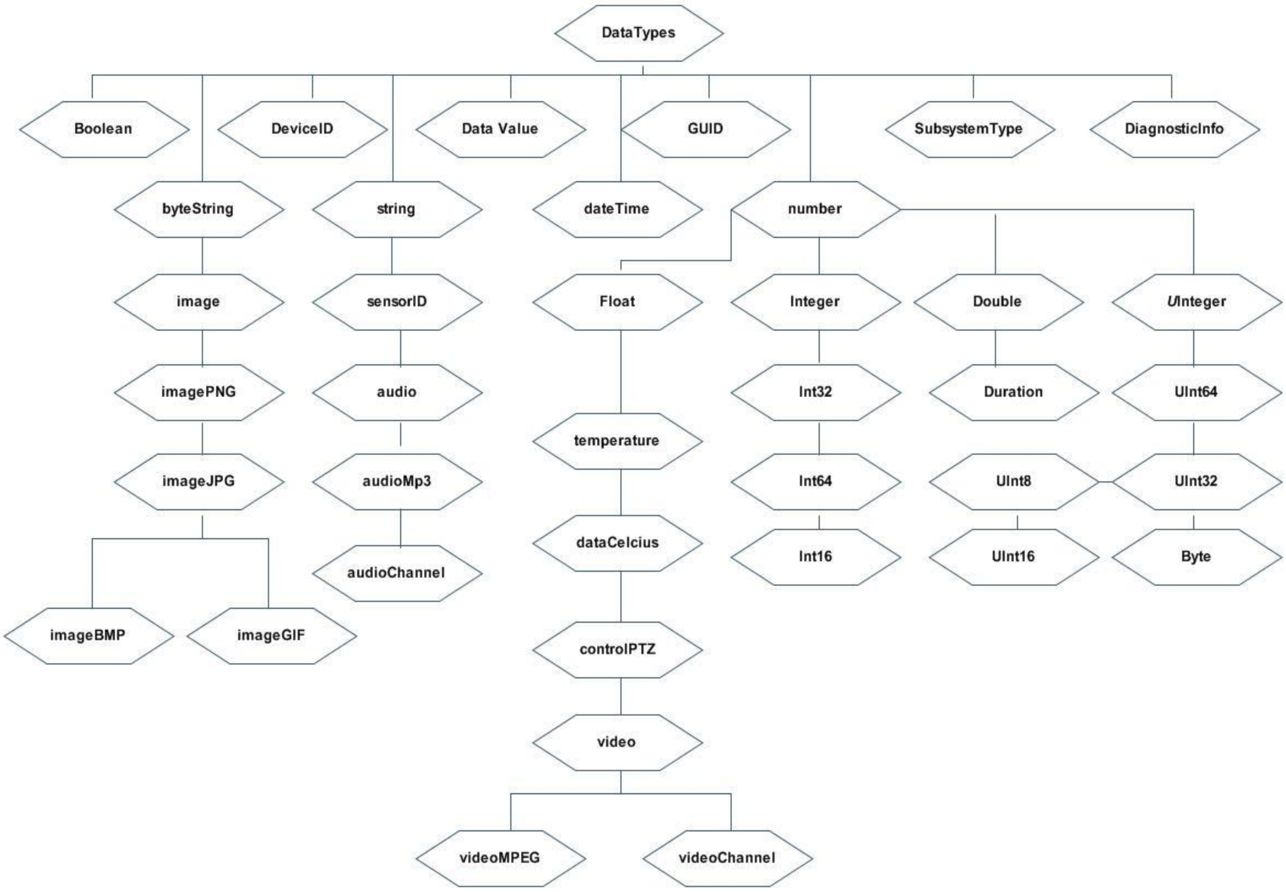

Figure 1: Classification of data type's hierarchy in intelligent building interoperation

There have been numerous studies proposed to evaluate the feature subset selection to obtain significant features in a given data (Glass and Cooper 1965, Hall 1999, Guyon, Weston et al. 2002, Ooi and Tan 2003, Hruschka, Hruschka et al. 2004, Jiang, Deng et al. 2004, Jirapech-Umpai and Aitken 2005, Díaz-Uriarte and De Andres 2006, Jafari and Azuaje 2006, Ma, Song et al. 2007, Rau, Jaffrézic et al. 2010, Yang, Zhou et al. 2010), and to solve the feature subset selection bias problem (Zhang 1992, Jensen and Neville 2002, Singhi and Liu 2006). To date, none of this FSAs has been able to identify the best combination of significant features that works efficiently as well as solving the challenge cause by the feature subset selection bias.

In this paper, a new framework, Multiple feature evaluation system (MFES) is proposed that uses a hybrid mechanism to tackle the problems of feature subset selection and subset bias problem of smart home data (Alemdar, Ertan et al. 2013). The mechanism uses the effectiveness of filters method with the accuracy of wrappers method. This paper is organized as follows: In section 2, it addresses related work on feature selection algorithms. Section 3 describes the proposed framework and its operation. Section 4 explains performance evaluation and we conclude in Section 5.

2 Related works

Several studies were proposed to address the feature subset selection and subset bias problems. In this section, it explains the solutions provided relating to these mentioned problems in machine learning paradigm.

Backstrom and Caruana conducted a work on cascade correlation using an internal wrapper feature selection technique. In this method, the features are selected while the hidden units are added to the growing correlation network architecture (Backstrom and Caruana 2006).

Liu et al conducted a work on wrapper method that utilizes the SVM model for demand forecasting. Firstly, a genetic algorithm based on wrappers is deployed to analyse the data. Then, the SVM regression model is then built by applying the selected data (Liu, Yin et al. 2008).

Deisy et al. conducted a research on filter method by using the analysis of symmetrical uncertainty with information gain (Deisy, Subbulakshmi et al. 2007). In their work, they calculated the difference between the features and the entropy of the whole class, which identifies the features that have less information. In addition, it highlighted that some methods of feature selection are based on features' discrimination ability.

Chen and Lin adopted the *f*-score technique for performing feature selection. In this research, the Support Vector Machine (SVM) was utilized for measuring the performance of feature sets. The *f*-score analyses each feature's decimation ability. Inferable from the SVM additionally attempts to find a separation hyper-plane to divide the different portions of the classes' data apart. The *f*-score helps the model in removing the features that have low decimation ability (Chen and Lin 2006).

Another important work was conducted on feature selection bias in regression. In their research, they focused on the ability to make inference in the built model with feature selection bias (Zhang 1992, Chatfield 1995). The highlighted work can make relationships between the output and the selected features stronger. It can negatively affect the prediction performance of the model, and that is because the model overfits the data in the presence of the selection bias and it may not generalize well. As a result of this findings, it has motivated the work. It is well known that classification was adopted for making qualitative predictions, while regression analysis was adopted for making quantitative predictions. However, results obtained from classification cannot be directly generalized to that of the regression.

In the work presented by Jensen & Neville, they used the context relational learning for feature selection bias. This is to indicate that some features may have an artificial linkage with the class, which causes their selection because of the relational nature of the data. Moreover, their work has been the first in terms of studying feature subset selection bias in classification. They used both independently and identical samples in parallel (Jensen and Neville 2002).

Surendra and Huan conducted an experiment using classification learning for the feature subset selection bias. In this method, a multi-class synthetic dataset was used for the experiment. Different feature selection techniques like IG, One-R, Chi-Squared, Relief-*f*, are applied to evaluate effects of the feature subset selection bias in classification task. In this experiment, it was observed that, an increase in number of instances frequently decreases the selection bias and, bigger attribute variance usually leads to bigger selection bias. The result shows that selection bias has less effect on classification performance than it does on regression performance (Singhi and Liu 2006).

Shaohua et al. [25] conducted a work on feature selection methods in machine learning. They proposed a novel immune clonal genetic algorithm to solve the problem of feature selection. In their experiment, they combined immune clonal and genetic algorithm to perform the experiment and predict the result. The algorithm performed better when compared to the genetic algorithm and the Support vector machine. However, the model eliminated the local search step for optimizing the features and, it did not cater for the feature bias problem.

Another work was conducted by (Huang, Cai et al. 2007). In their work, they proposed a hybrid genetic algorithm for feature selection using the wrapper method based on mutual information. The method uses learning machine as fitness function and searches best subset features in the space of all feature subset in the domain. It uses a heuristic algorithm to improve the local search for feature selection. The results show some improvement in the hybridization of the algorithm. However, it was noted that framework inherits the weakness such as the long run time and computational complexity.

In this research, a Multiple Feature Evaluation System (MFES) which introduces a hybrid feature selection method of filters and wrapper algorithms for finding best feature subset and tackles the feature subset bias problem in classification is proposed.

3 Proposed MFES Framework

This section explains the design of the MFES. It further explains filter and wrapper methods and discusses the operations of the multiple algorithms used in the implementation of the system. As significant feature selection is crucial in obtaining a relatively strong accuracy result when analysing a large dataset, the combination of these two different methods will improve the process. The filters processes quickly, however, the results obtained are usually not acceptable. In addition, it takes longer processing time in wrapper method, but it results in high accuracy values (Hsu et al. 2011). The filters technique calculates the information from features; and because of that, the results of the feature selection highly depend on the measured information of the features. On the other hand, the wrapper technique uses the learning algorithm for making judgement; however, the classification result obtained is biased by the learning algorithm. Table 1 depicts the overall properties of the two techniques.

Table 1: Properties of the filters and the wrappers.

Entities	Filters	Wrappers
Classification Accuracy	Depends on	High
Processing Speed	Fast	Slow
Dependency on Learning Methods	No	Yes

3.1 Filter and Wrapper methods

From information hypothesis perspective, the feature set information may well be determined by different statistical measures and because of that, it has been the centre for every filter type of feature selection method. As seen in figure 2, there are three main phases of filters which are feature generation phase, the measurement phase, and learning algorithm test phase. At the feature generation phase, a set of features are generated which are labelled as feature subsets. In the measurement phase, the current feature set information is measured. If the obtained results match the stop criterion, then the process is terminated, else, the steps will be performed repeatedly (Hsu et al. 2011). However, the stopping criterion is the threshold of the measurement results. Finally, at the learning algorithm test phase, an algorithm like Naïve Bayes (NB) or Support Vector Machines (SVMs) is utilized to test the significance of selected features. Hence, the final feature set contains the most significant features.

Figure 3 shows the procedure of the wrapper method. The process is like the filter method except a learning algorithm is used to replace the measurement phase. This has been the reason wrappers perform slower than the filters. However, due to the learning algorithm introduced at the second phase, the wrappers produce a more promising features selection results and solves the problem of feature subset selection bias. As the stopping criterion, when the number of features reaches some predefined threshold, the process stops automatically.

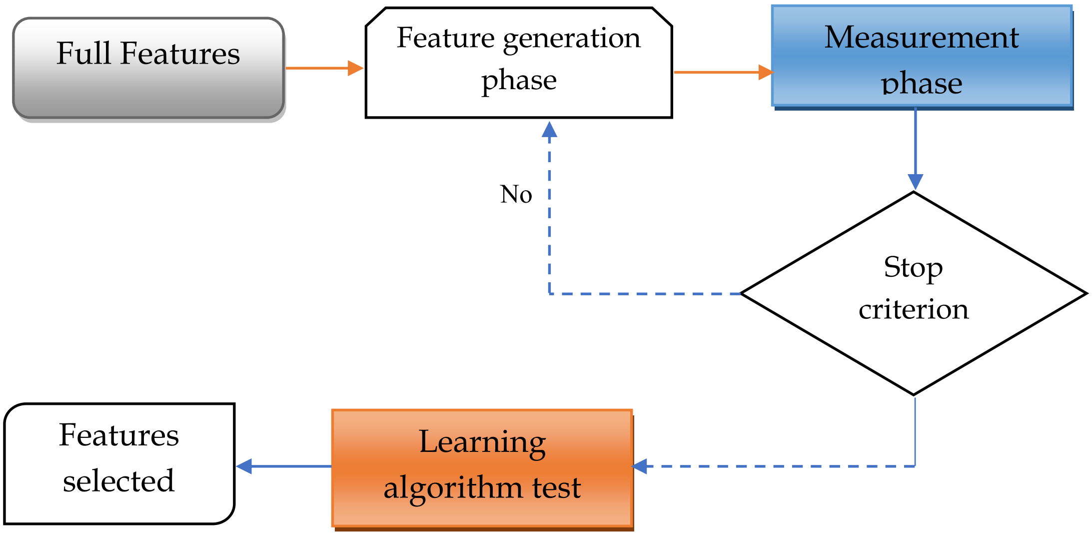

Figure 2: The filters method (f-score and Info Gain)

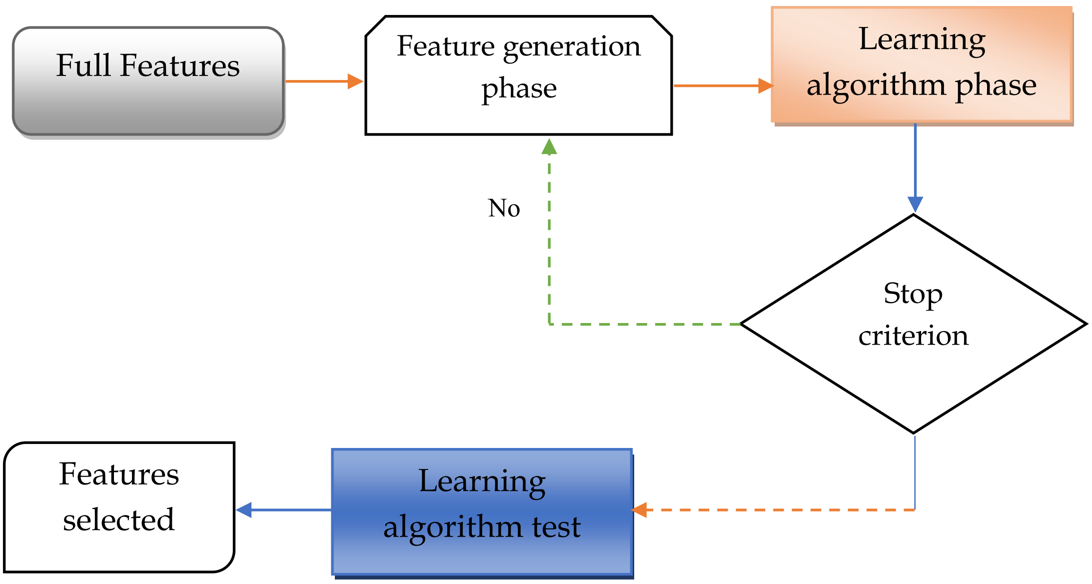

Figure 3: The wrappers method (Simulated Annealing Algorithm)

3.2 MFES (Hybrid) Framework

The overall MFES framework is illustrate in figure 4 below. It consists of mainly three stages: the preliminary screening stage where two filter methods (Information Gain and F-score) are chosen to remove/reduce most irrelevant or redundant features. These two resulted features are then combined at the second stage called Model combination stage. At the fine-tuning stage, a wrapper model with an experience of deep local search ability is then applied to eliminate the feature selection bias and improve the overall classification accuracy. The three stages are described in detail in the subsections.

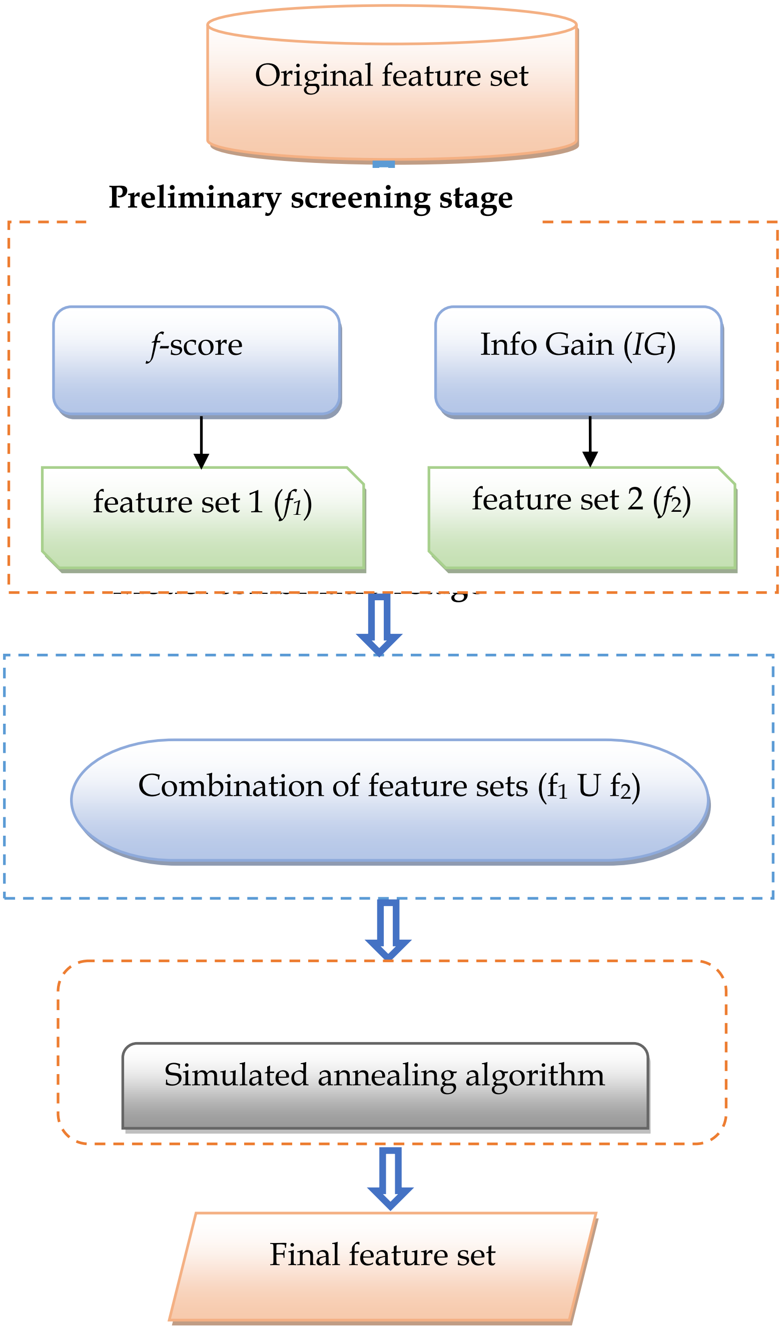

Figure 4: MFES framework

3.2.1 Preliminary Screening Stage

At this stage, both techniques which are the F-score, and IG were selected to reduce or remove irrelevant and redundant features of the IB dataset. F-score has been recognised as a novel model widely used in filters to calculate the discriminative capability of each feature, which means that in the classification problems, the features that have higher f-score have better separation capabilities. It can be illustrated as:

$$F(i) \equiv \frac{\left(\overline{x}_i^{(+)} - \overline{x_i}\right)^2 + \left(\overline{x}_i^{(-)} - \overline{x_i}\right)^2}{\frac{1}{n_+ - 1}\sum_{k=1}^{n+}\left(x_{k,i}^{(+)} - \overline{x}_i^{(+)}\right)^2 + \frac{1}{n_1 - 1}\sum_{k=1}^{n-}\left(x_{k,i}^{(-)} - \overline{x}_i^{(-)}\right)^2} \tag{1}$$

where $\overline{x_i}^{(+)}$, $x_i^{(-)}$ stands for the positive and negative averages of *i*th and x_i is the average of *i*th feature of the dataset; n_+ and n_ signifies the positive and negative number of instances respectively; and $x_{k,i}^{(+)}$, $x_{k,i}^{(-)}$ stands for *i*th feature of both *k*th positive and negative instances (Huang, Cai et al. 2007).

Consider equation 1, the larger the *F(i)* value is, the stronger the discriminative capability of the feature becomes. Since the F-score only examines the discriminative capability of only individual feature, it is not capable of identifying the multiple features. However, the features which score low are disregarded even if they can be complementary to the top features. Therefore, information gain (*IG)* which is another type of filter method was deployed which has the capacity to choose candidate features with related information.

$$Entropy(N) = \sum_{i=1}^{k} P_i \log_k \left(\frac{1}{P_i}\right) = -\sum_{i=1}^{k} P_i \log_k P_i \tag{2}$$

$$Entropy(D_j) = \sum_{i=1}^{|D_j|} \frac{D_{ji}}{N} \times Entropy(D_{ji}) \tag{3}$$

$$IG(D)_j = Entropy(N) - Entropy(D_j) \tag{4}$$

IG is concerned about the amount of information that can be provided by each feature. Equation 2-4 is used to determine the IG values. In equation 2, P_i stands for probability of class *i*, that appears in the *N* points of data, and it calculates all the classes information. D_{ji} is that the *j*th feature contains *i* kinds of different values as described in equation 3. Finally, the IG if the *j*th feature is calculated by finding the difference of equation 2 and 3.

3.2.2 Model Combination Stage

From the completion of the preliminary screening stage, feature sets F1 and F2 are selected when the two algorithms are run. The selected features obtained from the algorithms are classified as most class-related features (Huang, Cai et al. 2007). Considering the obtained features as the final set of features can be more harmful to the classification task and thus, it is not a wise decision. It will not only affect the computational time, but the classification accuracy will also be affected significantly and the key idea in this stage is how the two feature subsets obtained can be effectively combined. The combination of the two features sets i.e., the union (U) of the IG and F-score are separated into two distinct parts: exclusive-OR (XOR) and intersection (AND). Figure 5 illustrates the two parts of the selected feature set one (1) and two (2) respectively. From the full set, the intersection part of the feature sets (F1 and F2) is recommended by both algorithms (IG and F-score) and it can be conserved in our final feature set. However, for the exclusive-OR part of the feature sets (F1 and F2), some of the features might be included because they are valuable.

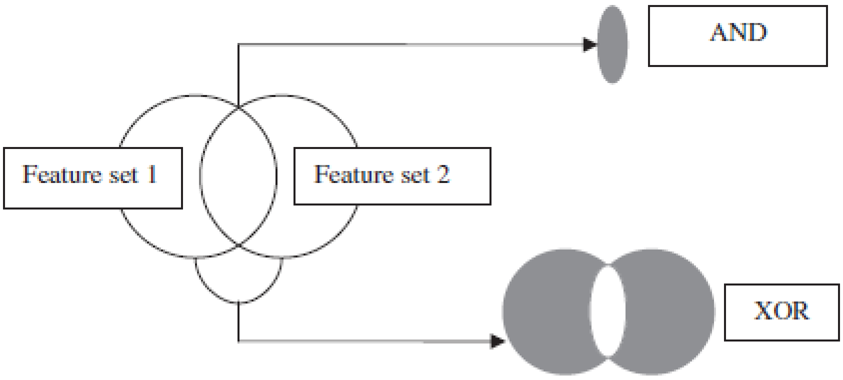

Figure 5: The Intersection and Exclusive-OR sets

Consequently, a fine-tuning stage which uses a wrapper method was implemented to further justify the significance of the selected features in both the intersection and exclusive-OR parts. However, the worst result coming from fine-tuning in respect to feature reduction is the union of feature set 1 and feature set 2.

3.2.3 Fine-tuning Stage (Simulated Annealing Algorithm)

At this last stage, a wrapper type of feature selection algorithm called simulated annealing algorithm (SAA) is used, which has a strong local search capability of selecting a feature set and can eliminates the feature bias problem while improving the classification accuracy. Since, the initialization of big data research in organizations, significant feature selection has become necessary for analysing meaning information. Thus, simulated annealing algorithm with the capability of deep searching has become a focal point for selection of significant features for big data analysis.

As discussed earlier, the wrapper method is not suitable when applied to wide range of features. Due to the reduction of feature performed at the preliminary screening stage, the wrappers can now be utilized with less computational effort. The simulated annealing solves the optimization problem by randomly manipulating the solution and then, increases the ration of greedy improvement slowly until it reaches a point where no further improvements are found (Kirkpatrick, Gelatt et al. 1983, Černý 1985).

To implement the simulated annealing technique, three parameters must be specified. First, the annealing schedule, consisting of the initial and final temperature, T_0 and T_{final}, and the annealing constant ΔT. This annealing schedule together govern how the search will proceed over time and when the search will stop. The second is the function, used for the evaluation of potential solutions (feature subsets). In this work, we assumed that higher evaluation scores are more significant. The neighbour function is the final parameter, it takes current solution and temperature as the initial input and returns new "nearby" solution. The temperature governs the size of the neighbourhood. At hight temperature, the neighbourhood becomes large and allows the algorithm to explore more and at low temperature, the neighbourhood becomes small, thereby forcing the algorithm to explore locally. Algorithm 1 describes the procedure for the simulated annealing for feature subset selection. The final features are obtained from this algorithm which is passed to any learning algorithm for evaluating the performance.

Algorithm 1: The Simulated annealing algorithm for feature subset selection

Given:

Examples **X** =< x_1, y_1>, . . . < x_m, y_m >

Annealing schedule, T_0, T_{final}, and ΔT with 0 < ΔT < 1

Feature set evaluation function *Eval* (., .)

Feature set neighbor function *Neighbor* (., .)

Algorithm:

S_{best} ← random feature subset

while $T_i > T_{final}$ **do**

Si ← *Neighbor* (S_{best} T_i)

ΔE ← *Eval* (S_{best}, **X**) − *Eval* (S_i , **X**)

if ΔE < 0 ***then*** *//if the new feature subset returns better*

S_{best} ← S_i

else *//if the new feature subset returns worse*

S_{best} ← S_i with probability exp $(\frac{-\Delta E}{T_i})$

T_{i+1} ← $\Delta T \times T_i$

return (S_{best})

Algorithm 1 optimizes the feature subset by iteratively improving the initial randomly generated solution. For every iteration, it generates a neighbouring solution and computes the difference between the candidate solutions and the current in terms of quality. It retains the new solution if it is better. Otherwise, it retains the new solution with a probability that is dependent on the quality difference, ΔE, and the temperature. The temperature is then reduced for the next iteration.

3.3 Training Model and Dataset

3.3.1 Training Model

In the fine-tuning stage, a more elaborate and critical machine learning algorithm was adopted to select the best set of features for classification. However, the significance of the selected feature set can only be tested using a classification algorithm in terms of some performance metrics. In our research paper, the multilayer perceptron (MLP) algorithm (Hastie, Tibshirani et al. 2009) which is a type of feedforward artificial neural network (ANN) to test the performance of the hybrid framework. The model is utilized because it adopts the property of a supervised learning technique for its training which is called backpropagation (George 2015) and has the capability of distinguishing data that are not linearly separable (Cybenko 1989).

3.3.2 Intelligent Building Datasets

We utilized the IB data (Alemdar, Ertan et al. 2013) for testing the MFES framework. The dataset is however collected in seconds, and for each day in the following format 00:00:00 to 23:59:59. It has several types of sensors used. In general, the database consists of about 28000 attributes observed in the building environment. To our work, to emphasize the performance of the framework, we focused on five sensors and measure the accuracy. Table 2 shows the dataset type and the sequential arrangement.

Table 2: Intelligent Building Dataset

Date	Time	Device/Action	Status	Device/Sensor ID
2014-02-27	12:43:27	Door	ON/OFF	5

2014-02-27	12:43:27	Energy management (EM)	ON/OFF	17
2014-02-27	12:43:27	Fire alarm	ON/OFF	6
2014-02-27	12:43:27	Public address	ON/OFF	24
2014-02-27	12:43:27	CCTV	ON/OFF	7

3.3.3 Experimental Setup

The experimental setup for the MFES framework was done with the following tools and environment.

- Windows 7-32bit operating system that runs on an Intel machine of corei7-3610QM.
- System specification of 20GB Hard disk space, 4GB RAM and 2.30GHz processor.
- Weka 3.8.0 tool and Notepad++ are deployed to run the feature selection algorithms.

4 Performance Evaluations

This section presents performance metric adopted to evaluate the proposed system. In addition, results and discussions are also presented.

4.1 Performance Metric

Accuracy: The Accuracy of prediction is used to characterize the classification algorithm's performance, derived as follows:

$$\mathbf{X} = \frac{\mathbf{Y}}{\mathbf{N}} * \mathbf{100}$$

Where:

Y stands for number of correct prediction and then N stands for total number of samples.

4.2 Results and Discussions

In the first step of the result, we used the two filter algorithms outlined in the preliminary screening stage to reduce the number of features. Table 3 depicts the results obtained from the preliminary screening process. In this method, a greedy process is used to resolve the threshold setting. The resulting accuracies of the two feature sets obtained from the two algorithms (F-score and IG) when run on MLP are 85.17% and 86.33% respectively. Also, the accuracy of the original feature set when run on the MLP resulted in 78.88%. Furthermore, the number of original features was however reduced to 18,554 from 28,668 for F-score and 19,004 for IG respectively. Nevertheless, with the reduction of features, the wrapper algorithm can perform much faster with little complexity.

Table 3: Preliminary screening procedure results

Algorithm	Threshold setting	No. of removed features	Retained (final) features	Performance metric (accuracy @10-fold cross validation) (%)
Unused	--	--	28668	78.88

F-score	0.001	10114	18554	85.17
IG	0.01	9664	19004	86.33

Table 4 describes the results obtained from the possible combinations of feature sets as discussed in section 3.2.2. We have the sum of 15790 features collected from the union set (F-score U IG), 10528 features obtained from the intersection set (F-score ∩ IG), and they are used in the fine-tuning stage (for running simulated annealing algorithm). The process determines which of the 5262 features obtained from the exclusive-OR set (F-score XOR IG) should be included. The accuracy results of both algorithms on the different parts (F-score U IG) and (F-score ∩ IG) are depicted. After taking the simulated annealing algorithm (SAA), a total of 6530 features are obtained and added successfully to the starting features obtained in the intersection part, resulting in a set of 17058 features in total and an accuracy of 92.17%. From the experiment, most features obtained from the preliminary screening stage are kept (F-score U IG). This mean the original feature set of 28668 is already very compact for this problem and most of the irrelevant features are removed by the preliminary screening stage. Furthermore, as obtained from the result, the accuracy performance has improved when the test is performed on the simulated annealing algorithm and the percentage (%) and number of features have reduced by about 45-48% (28668→17058). Consequently, that will certainly accelerate the subsequent process in this problem domain.

Table 4: Combination of Model results.

Association (relationship)	**No. of features**	**Performance metric (accuracy @10-fold cross validation) (%)**
The full feature set	28668	78.88
Feature (f-score U IG)	15790	85.1
Feature (f-score ∩IG)	10528	86.33
Feature (f-score XOR IG)	5262	--

Finally, to further elaborate on the proposed method, an existing method (Huang, Cai et al. 2007) obtained from literature is used to compare the results. To testing, same dataset was deployed for testing the performance of the framework. Table 5 compares the existing feature selection method with our proposed method. The result shows that our proposed method outperformed the existing method when applied to this domain.

Table 5: Comparison of proposed method with existing method.

Method	**Accuracy (%)**	**No of best features**
HGA [26]	87.21	14788
Proposed Method	92.17	17058

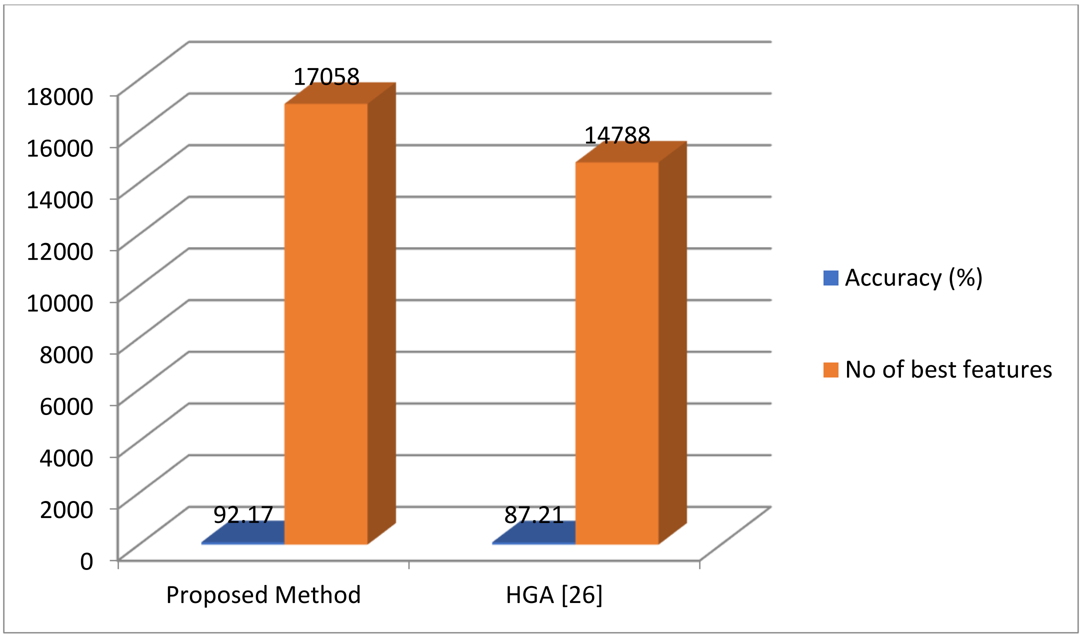

Figure 6: Comparison of Features and Accuracy results

5 Conclusion

A MFES framework was developed and tested in this research work. The idea is to find the most significant feature subset for the classification task. A 3-stage procedure was designed to have the best feature set in this problem domain. The preliminary screening is used to remove irrelevant and redundant feature. Feature bias is tackled at the fine-tuning stage and further examine the combined feature set. The results obtained showed that the hybridization of filters and wrappers yields to a better result in this problem domain.

References

Alemdar, H., et al. (2013). ARAS human activity datasets in multiple homes with multiple residents. 2013 7th International Conference on Pervasive Computing Technologies for Healthcare and Workshops, IEEE.

Backstrom, L. and R. Caruana (2006). C2FS: An algorithm for feature selection in cascade neural networks. The 2006 IEEE International Joint Conference on Neural Network Proceedings, IEEE.

Bellman, R. E. and S. E. Dreyfus (2015). Applied dynamic programming, Princeton university press.

Černý, V. (1985). "Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm." Journal of optimization theory and applications **45**(1): 41-51.

Chatfield, C. (1995). "Model uncertainty, data mining and statistical inference." Journal of the Royal Statistical Society: Series A (Statistics in Society) **158**(3): 419-444.

Chen, Y.-W. and C.-J. Lin (2006). Combining SVMs with various feature selection strategies. Feature extraction, Springer**:** 315-324.

Cybenko, G. (1989). "Approximation by superpositions of a sigmoidal function." Mathematics of control, signals and systems **2**(4): 303-314.

Deisy, C., et al. (2007). Efficient dimensionality reduction approaches for feature selection. International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007), IEEE.

Díaz-Uriarte, R. and S. A. De Andres (2006). "Gene selection and classification of microarray data using random forest." BMC bioinformatics **7**(1): 3.

George, F. H. (2015). Models of thinking, Psychology Press.

Glass, H. and L. Cooper (1965). "Sequential search: A method for solving constrained optimization problems." Journal of the ACM (JACM) **12**(1): 71-82.

Guyon, I. and A. Elisseeff (2003). "An introduction to variable and feature selection." Journal of machine learning research **3**(Mar): 1157-1182.

Guyon, I., et al. (2002). "Gene selection for cancer classification using support vector machines." Machine learning **46**(1-3): 389-422.

Hall, M. A. (1999). "Correlation-based feature selection for machine learning."

Hastie, T., et al. (2009). The elements of statistical learning: data mining, inference, and prediction, Springer Science & Business Media.

Hruschka, E. R., et al. (2004). Feature selection by Bayesian networks. Conference of the Canadian Society for Computational Studies of Intelligence, Springer.

Hsu, H., et al. (2011). "A hybrid feature selection by combining filters and wrappers " Expert Systems with Applications **38**(2011): 8144-8150.

Huang, J., et al. (2007). "A hybrid genetic algorithm for feature selection wrapper based on mutual information." Pattern Recognition Letters **28**(13): 1825-1844.

Jafari, P. and F. Azuaje (2006). "An assessment of recently published gene expression data analyses: reporting experimental design and statistical factors." BMC Medical Informatics and Decision Making **6**(1): 27.

Jensen, D. and J. Neville (2002). Linkage and autocorrelation cause feature selection bias in relational learning. ICML.

Jiang, H., et al. (2004). "Joint analysis of two microarray gene-expression data sets to select lung adenocarcinoma marker genes." BMC bioinformatics **5**(1): 81.

Jirapech-Umpai, T. and S. Aitken (2005). "Feature selection and classification for microarray data analysis: Evolutionary methods for identifying predictive genes." BMC bioinformatics **6**(1): 148.

Kirkpatrick, S., et al. (1983). "Optimization by simulated annealing." science **220**(4598): 671-680.

Liu, H. and L. Yu (2005). "Toward integrating feature selection algorithms for classification and clustering." IEEE Transactions on knowledge and data engineering **17**(4): 491-502.

Liu, Y., et al. (2008). Wrapper feature selection optimized SVM model for demand forecasting. 2008 The 9th International Conference for Young Computer Scientists, IEEE.

Ma, S., et al. (2007). "Supervised group Lasso with applications to microarray data analysis." BMC bioinformatics **8**(1): 60.

Ooi, C. and P. Tan (2003). "Genetic algorithms applied to multi-class prediction for the analysis of gene expression data." Bioinformatics **19**(1): 37-44.

Rau, A., et al. (2010). "An empirical Bayesian method for estimating biological networks from temporal microarray data." Statistical Applications in Genetics and Molecular Biology **9**(1).

Singhi, S. K. and H. Liu (2006). Feature subset selection bias for classification learning. Proceedings of the 23rd international conference on Machine learning.

Yang, P., et al. (2010). "A multi-filter enhanced genetic ensemble system for gene selection and sample classification of microarray data." BMC bioinformatics **11**(1): S5.

Zhang, P. (1992). "Inference after variable selection in linear regression models." Biometrika **79**(4): 741-746.