SELECTION OF A MINIMAL NUMBER OF SIGNIFICANT PORCINE SNPs BY AN INFORMATION GAIN AND GENETIC ALGORITHM HYBRID MODEL

Wanthanee Rathasamuth[1], ***Kitsuchart Pasupa***[2,*], ***Sissades Tongsima***[3]

[1,2]Faculty of Information Technology,
King Mongkut's Institute of Technology Ladkrabang, Bangkok 10520, Thailand

[3]National Center for Genetic Engineering and Biotechnology (BIOTEC),
National Science and Technology Development Agency (NSTDA), Pathum Thani 12120, Thailand

E-mail: rathasamuth.wan@gmail.com[1], kitsuchart@it.kmitl.ac.th[2*] (corresponding author), sissades@biotec.or.th[3]

DOI: https://doi.org/10.22452/mjcs.sp2019no2.5

ABSTRACT

A panel of a large number of common Single Nucleotide Polymorphisms (SNPs) distributed across an entire porcine genome has been widely used to represent genetic variability of pigs. With the advent of SNP-array technology, a genome-wide genetic profile of a specimen can be easily observed. Among the large number of such variations, there exists a much smaller subset of the SNP panel that could equally be used to correctly identify the corresponding breed. This work presents a SNP selection heuristic that can still be used effectively in the breed classification. The features were selected by combining a filter method and a wrapper method–information gain method and genetic algorithm–plus a feature frequency selection step, while classification used a support vector machine. We were able to reduce the number of significant SNPs to 0.86 % of the total number of SNPs in a swine dataset with 94.80 % classification accuracy.

Keywords: Bioinformatics, Feature selection, Information gain, Genetic algorithm, Support vector machine, Swine, Single nucleotide polymorphisms.

1.0 INTRODUCTION

Swine breed improvement has played an important role in boosting the quality and quantity of pork in the market. Examples of swine breeds that are currently popular in many countries are Landrace, LargeWhite, Duroc, Creole, Wild boar and Hampshire [1]. Each breed has distinctive characteristics. For example, the breeds that are commonly used as breeders are LargeWhite, Landrace, and Duroc because they are strong, grow quickly and provide a good quality carcass, especially the LargeWhite breed. The Duroc breed, on the other hand, grows well under any weather conditions and is very popular as a breeder for beautiful hybrids, while the Landrace breed is very good at rearing its offspring but carries poor traits, e.g., having weak legs. Therefore, cross breeding among these breeding stocks has become a common practice to produce desired characteristics.

The unique characteristics of each breed are manifest in differences in the deoxyribonucleic acid (DNA) base sequence of each breed. DNA is a nucleic acid that stores genetic information of living beings. Unfolded, DNA can be seen as an arrangement of several nucleotides sequentially connected into two intertwining strands of polynucleotides that consist of four kinds of bases: adenine (A), thymine (T), cytosine (C), and guanine (G). Base pairing between the two polynucleotide strands is a complementary base paring by hydrogen bonds: Adenine pairs with thymine and cytosine pairs with guanine. Two molecules of nucleotide can be arranged in DNA strands in 16 ways (or 4^n, where n is the number of molecules). Therefore, in a typical DNA molecule that has hundreds of thousands or even a million nucleotide pairs, the base sequence of the DNA molecules from two different individuals will be very different. It is the source of genetic polymorphism, such as skin color, height, severity of contracted disease and diverse responses to drugs. These diverse characteristics stem from different sequential base sequences, which is called single nucleotide polymorphism (SNP), that can occur at any of the million positions in a DNA chain. It has been estimated that SNP can be found in every sequence of 300 bases. An example of different base sequences that result in an SNP is between

GCAACGTTGA and GCAGCGTTGA. This SNP is found in more than 1 % of individuals in a population. It is just called point mutation, if it affects a smaller percentage of a population. Porcine SNP analysis can determine the SNPs that affect its reproduction and growth. However, since there are over one million SNPs in the DNA of a living organism, an SNP analysis by a human expert is out of the question, notwithstanding the cost and resources needed to do it. Therefore, a better way to address this issue is to apply bioinformatics which is an integration of biology and medical and computer sciences. Various techniques for processing data by computer have been adapted for uses in bioinformatics. One of the most powerful and developed computer techniques is machine learning, which integrates the tools of computer, engineering, and statistics. Broadly speaking, this technique enables a computer to respond to new data by itself based on prior information. Machine learning has been used in several branches of bioinformatics such as genomics, proteomics, microarray, systems biology, evolution and text mining [2]. This reference also describes several machine learning techniques such as Support Vector Machine (SVM), Bayesian classifiers, Decision Tree, k-Nearest Neighbors, and Artificial Neural Networks. Machine learning can be divided into three categories according to the type of learning: supervised learning, unsupervised learning and reinforcement learning. We used the supervised learning technique, which involves construction of a predictive model from a training dataset and validation of the model with a testing dataset. The algorithms used in supervised learning can be either regression or classification. In this study, it is a classification task.

In general, a learning technique for constructing a model can support a large number of features but often is not effective at classification due to over-fitting when there are more features than samples. Over-fitting occurs when the constructed model has too high an accuracy, which, when used with a test dataset gives a low prediction accuracy. One way to solve this uses a small number of features. Hence, several techniques for reducing the number of features have been reported in [3, 4]. These review papers report applications in bioinformatics, that have used feature selection techniques, such as taxonomy, microarray domain and mass spectrometry. They also report three types of feature selection techniques used in bioinformatics: 1) filter methods such as Euclidean distance, *i*-test and information gain (IG); 2) wrapper methods such as genetic algorithms (GA) and other nature-inspired algorithms; and 3) embedded methods such as Random Forest, SVM weight vector and Decision Trees.

The differences in feature selection by the filter, wrapper and embedded methods are described next. The filter method selects features by sorting feature indices and selecting the indices with the highest rank; feature selection and classification are independent of each other here. The advantages are that it is simple and fast. The wrapper method selects features by evaluating the suitability of each subset of features after classification, resulting in a subset of features that can give high classification accuracy. Since evaluation follows classification, a large number of features need a long computation time. The embedded method is very similar to the wrapper method; but in an embedded method, features are selected concurrently with classification model construction and hence uses less computation time than a wrapper method. Since both wrapper and embedded methods use classification to select a subset of features, they provide good feature learning and give good prediction accuracy with a training dataset. However, this good accuracy may come at the expense of over-fitting.

Wrapper methods have been widely used for feature selection, especially various nature-inspired algorithms, for example, used by Zang, Zhang and Hapeshi [5] with the objectives of increasing the efficiency and reducing the prediction error. Chuang et al modified a particle swarm optimization technique to design an Improved Binary Particle Swarm Optimization (IBPSO) and for selecting gene expressions in combination with a k-Nearest Neighbors classifier. IBPSO avoided getting trapped in local optima and gave good classification results [6]. Huang [7] designed a new classifier model, a hybrid Ant Colony Optimization based classifier model, that integrated Ant Colony Optimization techniques with SVM in order to improve classification accuracy by using a small number of discriminating features. There are also several studies used nature-inspired techniques for bioinformatics: Nakamura et al's bat algorithm [8], Rodrigues et al's Cuckoo search algorithm [9] and Flower Pollination algorithm [10] that proposed using Bat algorithm, in combination with an optimum-path forest classifier. GA has been applied for pattern recognition [11, 12], investigation of protein function [13] and SNP selection [14]. Peng et al [15] and Li et al [16, 17] used GA with SVM in bioinformatics. Lei [18] combined GA with IG to achieve better classification accuracy, than each of the technique alone.

Since the swine data used here, consisted of a large number of SNPs, the number of swine samples was small and it was expected that only a few SNPs would affect the classification, we aimed to find and select the smallest number of SNPs, that led to effective classification. Here, we selected features with a hybrid IG+GA, a fast filter method combined with a

random and selective wrapper method, which includes an SNP selection step based on the frequency of appearances in randomly-seeded datasets constructed from the whole dataset. We hypothesized that the most relevant SNPs should be the ones that appear in most of the randomly-seeded datasets. The rest of the paper is arranged as follows: section 2.0 describes the methodology, including feature selection and classification techniques; section 3.0 describes the dataset used; section 4.0 describes the experimental setup; section 5.0 describes and discusses the results and we conclude in section 6.0.

2.0 METHODOLOGY

In this section describes the conceptual framework and feature selection by a hybrid IG and GA (IG+GA) technique. This technique assumes that many features of the full feature set are not significant in constructing a learning model, but waste computer resources and lengthen computation time. The technique was intended to select the minimum number of significant features, that can classify SNPs accurately. IG, GA, IG+GA, SVM (the classifier) are explained briefly, along with feature selection according to their frequency of appearance in randomly-seeded datasets.

2.1 Information gain

IG is a feature selection technique of in filter method class [3, 4, 19], that selects features according to ranked index weights, calculated from the relationships between features. It is a common feature reduction technique, that can boost classification capability of any classifiers. It has been applied to applications, such as feature selection in text, DNA microarrays and SNPs. Cho and Won [20] used it to select features in a DNA microarray: they showed that IG was the best among all the techniques tested, including Multi-layer Perceptron and k-Nearest Neighbors. Jirapech-Umpai and Aitken [21] used six filter methods to rank features and three cut-point determination methods to find a good cut-point: they found that IG was the best feature ranking method and Z-score analysis was the best cut-point determination method. Using these two methods in combination, the microarray was classified with the highest accuracy.

The IG value of each feature was calculated from the difference between the initial and current information entropy of the feature. Entropy is a measure of unpredictability of the state, or equivalently, of its average information content. An information entropy signifies the difference between data points: a higher entropy means that the data points are very much different, while a lower entropy means that the data points were not very different. Therefore, a feature with a high IG value is a good feature. Calculation of IG value is expressed in Eq. (1) below,

$$IG(T,i) = H(T) - \sum_{v \in vals(i)} \frac{|\{\boldsymbol{x} \in T | x_i = v\}|}{|T|} \cdot H(\{\boldsymbol{x} \in T | x_i = v\}), \quad (1)$$

and H is information entropy that can be calculated by

$$H(T) = -\sum_{x \in \boldsymbol{x}} p(x) log_2\, p(x) \quad (2)$$

where T is training dataset with samples in the form of $(\boldsymbol{x}, y) = \{x_1, x_2, \ldots, x_k, y\}$where $x_i$is the feature at the present position i of the sample and y is the corresponding class label of the $\boldsymbol{x}$ sample; and p(x) is the proportion of the number of elements in sample $\boldsymbol{x}$ to the number of elements in set T.

IG can rank features according to their significance but cannot determine the optimum number of features for classification. We used an elbow method to reliably determine the cut-point, i.e. the number of highest-ranked features sorted by IG, that would be optimum for classification. The elbow method is a method of interpretation and validation of consistency within cluster analysis designed to help finding the appropriate number of clusters in a dataset.

2.2 Genetic algorithm

GAs are nature-inspired algorithms. As a feature selection technique, they fall into the wrapper method category. GAs mimic evolution in nature and genetic inheritance in its search for an optimum solution. It crosses over solutions, then selects better solutions, represented by chromosomes, that contain several genes. In GAs, chromosomes are in the form of strings of alphabets or binary bits. In recent years, GAs have been used for reducing the number of data dimensions in pattern recognition [11, 12]. As mentioned above, for feature selection process, too many features, but too small a

number of samples, will degrade machine learning performance. Many studies on feature selection have used various techniques to solve this issue [5, 6, 7]. GAs have been widely used for feature selection [15, 16, 17]. They were combined with SVM and used in a bioinformatic application - classification of array-based multiclass tumor [15]. They have also been used for predicting protein function [13], where a GA was used to select some variables before they were used further by SVM. Their prediction results were compared to those of Borro et al. [22] and found to be clearly better, demonstrating that using a GA to select a small number of significant variables was more effective, than Borro et al.'s technique. İlhan et al. [23] used a GA to find the optimum parameters, including hyper-parameters, for SVM operation.
A GA specifies the number of chromosomes with their gene components in the population, specifies their fitness function for the evolution process, generates a random initial population, applies genetic operators–selection, crossover, and mutation–to the population, then repeats these steps to form a new population, until the stopping criterion is met. Fig. 1 shows the steps of a GA - specifically for feature selection. These steps are explained in detail below.

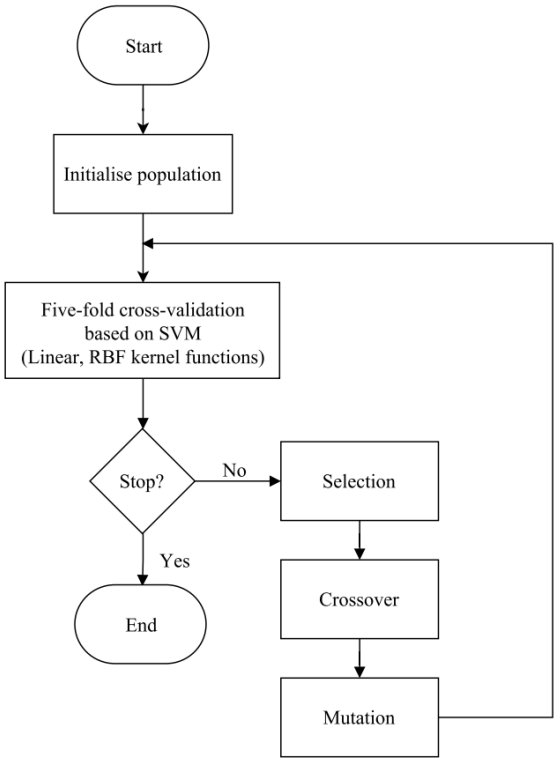


Fig. 1: Steps for a GA in operation with SVM

1. Generate an initial random population of chromosomes, that are binary bit strings: Each chromosome $\boldsymbol{s}$ consists of n genes s, where $\boldsymbol{s} = \{s_1, s_2, \ldots, s_n\}$. Fig. 2 shows an example of chromosomes represented by bit strings. s_1 and s_2 are two chromosomes, each containing 10 binary bit strings of genes (n = 10), that represent 10 features in the sense that a binary 1 at a position in the string means that the corresponding feature in the training dataset is selected for fitness function evaluation to find out which chromosome is the best one. In this study, the fitness function was accuracy. In Fig. 2, s_1 and s_2 have three and five selected SNPs, respectively.

2. Select to-be-reproduced chromosomes by roulette wheel method: This method selects a chromosome randomly, based on its selection probability, which is the ratio of its fitness to the total fitness of the entire population.

3. Crossover of two chromosomes: exchanges some of their genes to form new chromosomes, that may be better than the original ones. Crossover is a multi-point crossover, that starts by generating random numbers, that specify the positions and blocks of genes that will be crossed over.

4. Mutation of the cross-over chromosomes: mutation increases the diversity of a chromosome population. Even though the selection and crossover operators may give better solutions, the solutions are still based on the original chromosomes and so may not be diverse enough to reach a global optimum. Mutation can generate diverse solutions, that may not be obtainable from information stored in the parent chromosomes. The first mutation method used in this study was bit-flip mutation - used in the original GA formulation. Bit flipping is based on a mutation probability, P_m. For instance, $P_m = 0.01$ implies that the bit representing the gene has a 1 % chance to flip from 0 to 1 or 1 to 0.

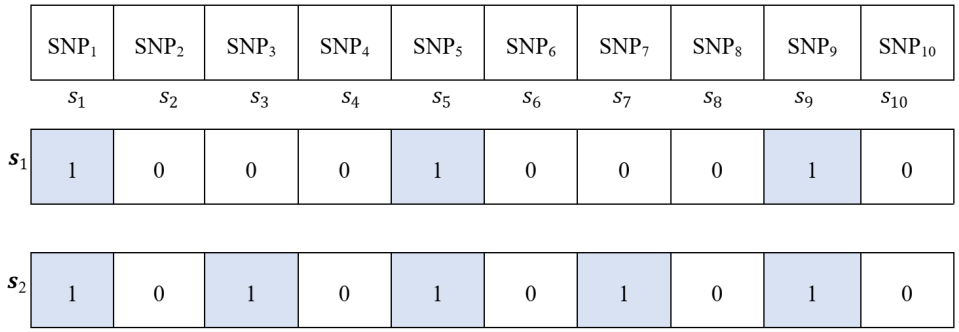


Fig. 2: Example of binary bit strings of genes that make up two chromosomes

However, it was found that after mutation, the number of 1s in each mutated chromosome was still too high, 50 % of all genes, and this could generate too many eligible features. To obtain a smaller number of optimum features, we used a higher probability for 1 to 0 bit-flipping than that for 0 to 1 bit-flipping, as shown in Eq. (3) [14], which was shown to be successful. For example, $P_m = 0.1$ implies that the bit representing that gene has a 10 % chance to flip from 0 to 1 and 1 to 0 was much higher at 90 %.

$$s(i) = \begin{cases} 1, & r \leq P_m \\ 0, & Otherwise \end{cases} \tag{3}$$

Eq. 3 includes the conditions for flipping existing bits, where $s(i)$ is the flipped bit at i, and r is a random number $\in [0, 1]$

2.3 Information gain and genetic algorithm hybrid

Lei [18] used IG+GA for text classification. Information gain calculated how many terms can be used for the classification of information in order to measure the importance of the lexical items for classification. Subsequently, GA was used to select the most suitable features.

GA alone could not reduce the number of features sufficiently in the SNP-feature-reduction tests that we ran. Even though IG+GA could reduce the number of features to a minimum, those features did not result in accurate predictions due to an insufficient number of features. Consequently, we chose to employ a different approach for combining IG+GA for our classification task, where IG was used to rank features according to their significance. An elbow method was used to find a cut-point for inclusion of only some of the features obtained from IG, which also specified the number of genes in each chromosome in subsequent Proposed GA. GA was used to further reduce the number of these features down to a suitable number by adjusting the mutation probabilities for 0 to 1 bit-flipping and 1 to 0 bit-flipping separately. A suitable number of features here means that they provided good classification accuracy in test runs.

2.4 Support vector machine

SVM is a machine learning technique for the supervised learning category. It was developed to solve binary classification problems. The main concept of this technique is hyperplane construction. In SVM, a hyperplane is a decision plane for dividing data into two classes. An optimum hyperplane has the largest margin between the two classes. The data on the margin are called support vectors. SVM can have one of many kernel functions such as linear,

radial basis function (RBF), and polynomial kernels. These different functions map data from input space to feature space with higher dimensions. Each kernel function is appropriate for a different kind of problem: the function used does not always need to be linear, depending on the type and complexity of the input data. SVM has been applied as a classifier in several research studies [15, 16, 17]. In this study, linear and RBF were tested, and their performances compared. For the test, C is a hyperparameter of SVM that balances training error and the model's complexity. Particularly for the RBF kernel, a parameter γ was tuned to get the optimal hyperplane. The optimal parameters were validated using a five-fold cross-validation procedure. The mathematical expressions for the linear kernel function and RBF are in Eq. (4) and (5), respectively.

$$k(x, x') = x^T \cdot x' \tag{4}$$

$$k(x, x') = \exp(-\gamma \|x - x'\|^2) \tag{5}$$

where $k(x, x')$ is a kernel function; x and x' are data samples; the term $\|x - x'\|^2$ is a squared Euclidean distance between x and x'; and γ is a non-negative value.

A diagram of the feature selection by our proposed approach is shown in Fig. 3. In this approach, we also compared the effectiveness of IG and GA alone as well as of IG+GA. Accordingly, their uses are as shown in the diagram. As mentioned in the section above, we introduced different flipping probabilities of 1 and 0 into GA. Therefore, from now on we will call this GA the "Proposed GA". Please note that both GA and Proposed GA were used individually and in combination with IG. This is not shown explicitly in the diagram.

As can be seen in the diagram, pre-processed data were initially divided into two sets for this experiment: training and test sets. The training dataset was used in feature selection. In our experimental framework, three feature selection methods were evaluated: filter, wrapper, and a combination of filter and wrapper.

1. The filter method used IG for ranking the level of significance for each feature, while an elbow method was used to find the cut point for selection of the optimal number of features.

2. The wrapper method used GA for selection of an optimal subset of features for classification. To find this optimal subset, GA needed to send a preliminary subset of features into the classification process used for training, testing to find optimal parameters and evaluating the SVM model by five-fold cross validation. The best subset of features gave the highest prediction accuracy.

3. The filter plus wrapper combination method performed the filter and wrapper methods in that order. The cut point from the elbow method in the filter method would set the number of genes in each chromosome to be performed in GA.

In the development of our approach, we made an assumption that, from all 10 randomly-seeded datasets, it was likely that some features from every dataset would be repeatedly selected. Thus, the high frequency of occurrences in the selected features meant that they were the most significant features. Therefore, we introduced a feature frequency selection (FFS) step after the selected features from IG, IG+GA, and IG+Proposed GA were obtained in order to select only a small number of the most significant features. Briefly, FFS works to find the previously selected features that have the highest frequency of occurrences among randomly-seeded datasets. In this study, FFS was performed separately on the features selected by each of these kernels since IG+GA+FFS and IG+Proposed GA+FFS used both linear and RBF kernels. The newly selected features from both kernels were then combined and the same ones were taken as the finally-selected features, as shown in Fig. 4.

After features were selected, they were used in a training step, through five-fold cross-validation, to find the optimal parameters for constructing the optimal model. Specifically, for RBF kernel in SVM, the grid search method is used to find the best C and γ for the SVM model. The optimal parameters are inserted into the prediction step together with the test dataset. The output is classification accuracy.

3.0 DATASET

All swine data used in this study was from the Porcine Colonization of the Americas Dataset [1]. It consists of data from 11 village pig breeds including the Creole, Moura, Yucatan, Ossabaw pig, Monterio, and Guinea hog, which are raised in the United States of America, as well as 10 outgroup pig breeds including Jiangquhai, Jinhua, Meishan, Xiang pig, Duroc, Landrace, and LargeWhite. The dataset contains data from 389 pig samples and 46,259 SNPs, which was gleaned by a PLINK method from 62,163 SNPs. Some of the breeds presented in the dataset had too few samples representing them. Consequently, those breeds were excluded from the study. In total, the dataset that we used in this study consisted of data from 356 samples of 21 breeds, as shown in Table 1, and a total of 16,579 SNPs. All data were put through data cleansing according to the principle of population and sample identification. However, there were some missing values. Thus, they were estimated by a single imputation method. The estimated values were modes of the entire individual feature data.

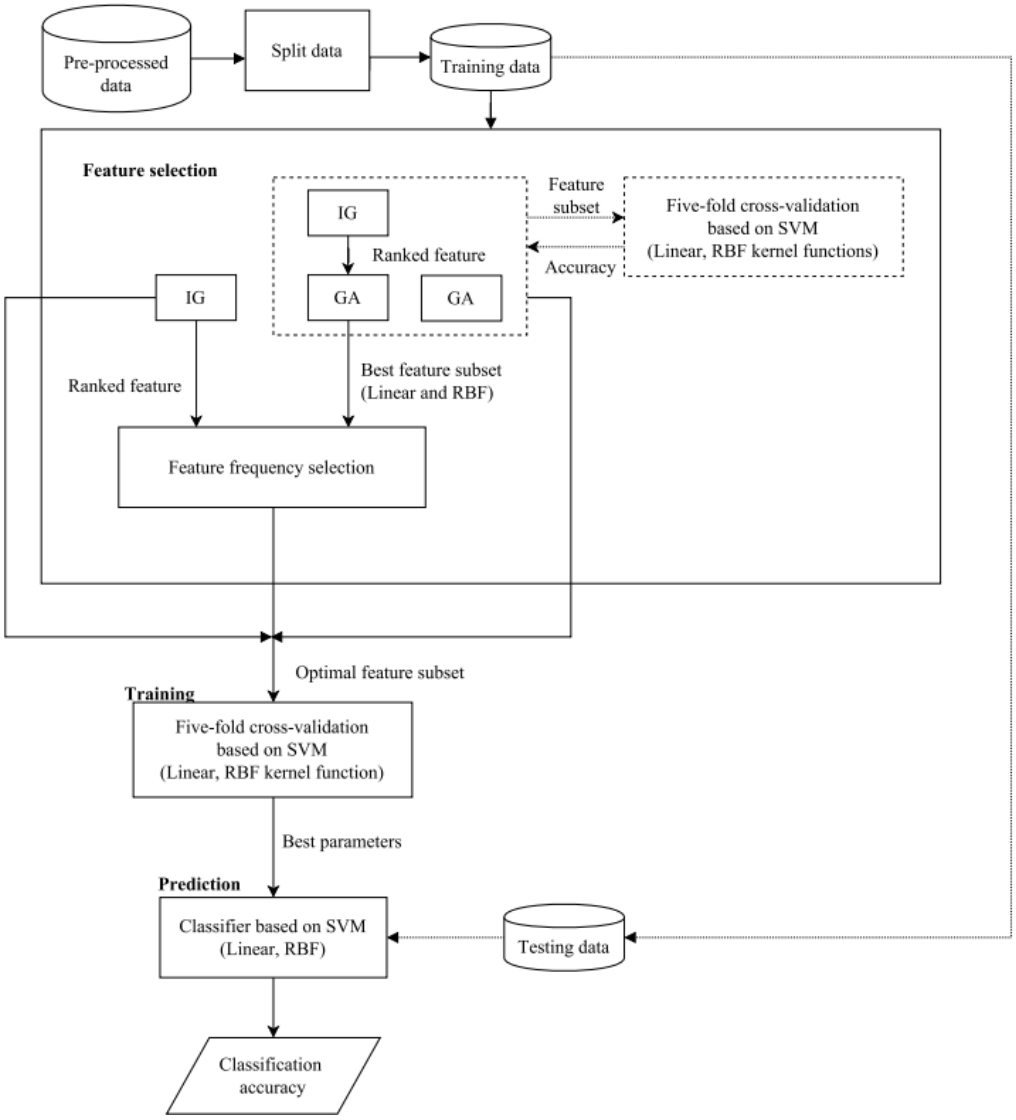


Fig. 3: Experimental framework of the feature selection for classification

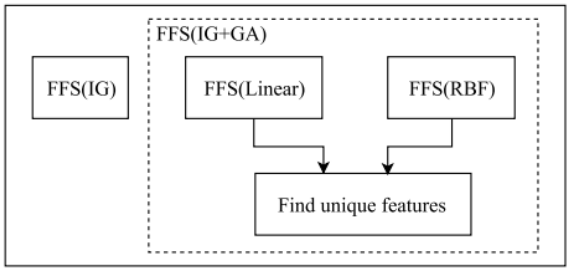


Fig. 4: Application of FFS for combining and selecting features from linear and RBF kernels

Table 1: An example of swine data in the dataset used in this study

Breed	**Location**	**Number of samples**
Creole	Alto Baudo-Colombia, Baja Verapaz-Guatemala, Granma Cuba, Guanacaste, Alajuela-Costa Rica, Loja-Ecuador, Misiones Argentina, Pinar del Rio-Cuba, Titicaca area-Peru	90
Piau	Bahia-Brazil	9
Zungo	Cerete-Colombia	10
Jiangquhai	China	11
Jinhua	China	16
Meishan	China	16
Xiang pig	China	11
Moura	Concordia-Brazil	9
Duroc	Denmark, Holland, USA	20
Landrace	Denmark, Holland, USA	20
LargeWhite	Denmark, Holland, USA	20
Semi- feral	Formosa-Argentina	10
Wild boar	Hungary, Poland, Tunisia	13
Yucatan	Indiana-USA	10
Hairless	Mexico	9
Cuino	Nayarit-Maxico	7
Ossabaw pig	Ossabaw island-USA	7
Monteiro	Pocone-Brazil, Portugal	24
Iberian	Spain	15
Hampshire	UK, USA	14
Guinea hog	USA	15

4.0 EXPERIMENTAL SETUP

The entire swine dataset was used to construct 10 randomly-seeded datasets. This large number of randomly-seeded datasets was used to make the results of this experiment statistically valid and reliable. Each randomly-seeded dataset was partitioned into a training dataset (80 %) and a testing dataset (20 %). The parameter settings of GA, Proposed GA, IG+GA, and IG+Proposed GA were as follows: population size of 30 chromosomes; crossover probability of 0.8; P_m ranging from 0.1 to 0.9; the number of genes of 16,579 for GA and Proposed GA; the number of generations of 10; C ranging from 10^{-6} to 10^{6}; and γ of RBF ranging from 10^{-10} to 10^{10}. In FFS for IG, IG+GA, and IG+Proposed GA, features with frequency 80 % and higher were selected, i.e. features that occurred more than or equal to eight times in the 10 randomly-seeded datasets.

The reason that we set the population size to a small number of 30 was that a higher number would result in a large number of features which would have wasted significant amount of computational time. In addition, the reason that we set the number of generations to 10 was that the preliminary trial runs showed that GA met its stop criteria within 10 generations. Thus, setting it to a higher number was not likely to increase the accuracy in any way. The full experimental results are reported in Section 5.0 below.

5.0 EXPERIMENTAL RESULTS AND DISCUSSION

In this section the result of SNPs selection for classification are presented and discussed. Besides presenting the average accuracy of classification by the proposed approach and the numbers of finally–selected SNPs by every method used in this approach, we also present the average accuracy of classification from using all the features for comparison. All of these results are displayed in Fig. 5 and 6. This section also presents the statistical results of ANOVA analysis of the prediction accuracy achieved by every method and the results of principal component analysis (PCA).

In our Proposed GA, mutation probability refers to the frequency of new mutations per generation in an organism or a population. It was observed that from the range of P_m (0.1-0.9) set for running GA, Proposed GA, IG+GA, and IG+Proposed GA, the values of the best-tuned P_m for those methods were 0.8, 0.2, 0.5, and 0.8, respectively, as shown in Fig. 5, for each value of the P_m tested, GA, IG+GA, and IG+Proposed GA gave nearly the same number of selected SNPs, as shown in Fig. 5a, 5c and 5d, so the optimal value of P_m was considered to be the value that provided the highest classification accuracy. On the other hand, for Proposed GA, the number of selected SNPs obtained from using different values of P_m was not nearly the same as shown in Fig. 5b, but the number of selected SNPs from P_m of 0.1 and 0.2 was the lowest and nearly the same. Classification accuracy from P_m of 0.2 was higher than that from P_m of 0.1, so it was used as the optimal P_m for Proposed GA. To conclude, it can be seen that for GA and IG+GA that used the original type of bit-flip probability, no matter what P_m value was used, the total number of selected SNPs was large, about 50 % of search space. For Proposed GA and IG+Proposed GA that used our proposed type of bit-flip probability, however, the total number of selected SNPs was smaller, though the optimum number depended on the size of the search space; a large search space needed a smaller value for P_m, but a small search space needed a larger value for P_m. The results presented here are from using these tuned values of P_m with the respective methods.

The highest levels of classification accuracy from the training step of GA, Proposed GA, IG+GA and IG+Proposed GA are shown in Fig. 8. It can be seen that the number of generations at the stop of a run for the first set of randomly-seeded dataset of each tested method was not over 10. In addition, IG+GA and IG+Proposed GA gave better levels of accuracy than GA and Proposed GA alone. The average number of generations at the stop of runs for all 10 randomly-seeded datasets of each tested method was between 3 and 7 generations, as shown in Fig. 9.

5.1 Classification accuracy and number of selected SNPs

The resulting average classification accuracies and the number of selected SNPs are summarized in Table 2. It can be seen that all of the methods used were competitive. The best method was IG+Proposed GA+FFS, which provided a classification accuracy of 94.62 % and 94.80 % for linear and RBF kernels, respectively, showing that the proposed approach was able to achieve a better classification accuracy to that when using the entire features from the dataset (92.46 %) while using far fewer features–only 0.86 % of SNPs. The worst method was IG+GA+FFS, which exhibited a classification accuracy of 76.62 % for linear kernel and 77.08 % for RBF kernel, markedly lower than any other methods. The reason for this might be that it provided too small a number of SNPs (21 SNPs) to be able to make an effective classification. It can also be seen that the number of SNPs selected by wrapper methods, GA and Proposed GA, were still too high, 49.70 % and 7.09 % in linear case, respectively. When a filter method, IG, was used in combination with the wrapper method, the number of selected SNPs reduced dramatically. For instance, IG+GA and IG+Proposed GA were able to reduce the number of features to 1.00 % and 1.44 % of the entire features in linear case, while IG alone was able to reduce it to 1.98 %. However, using too small a number of SNPs could lead to drop in performance, as in the IG+GA case. When FFS was added, the numbers of selected SNPs were further reduced: IG+FFS, IG+GA+FFS, and IG+Proposed GA+FFS were able to reduce the number of selected SNPs to 1.22 %, 0.31 %, and 0.86 % of the entire SNPs in the dataset, respectively. Fortunately, accuracy could be improved in most cases, except for IG+GA+FSS case–using too small a number of features (21 SNPs). The best performer was IG+Proposed GA+FFS, which was able to select only 142 SNPs from the total of 16,579 SNPs.

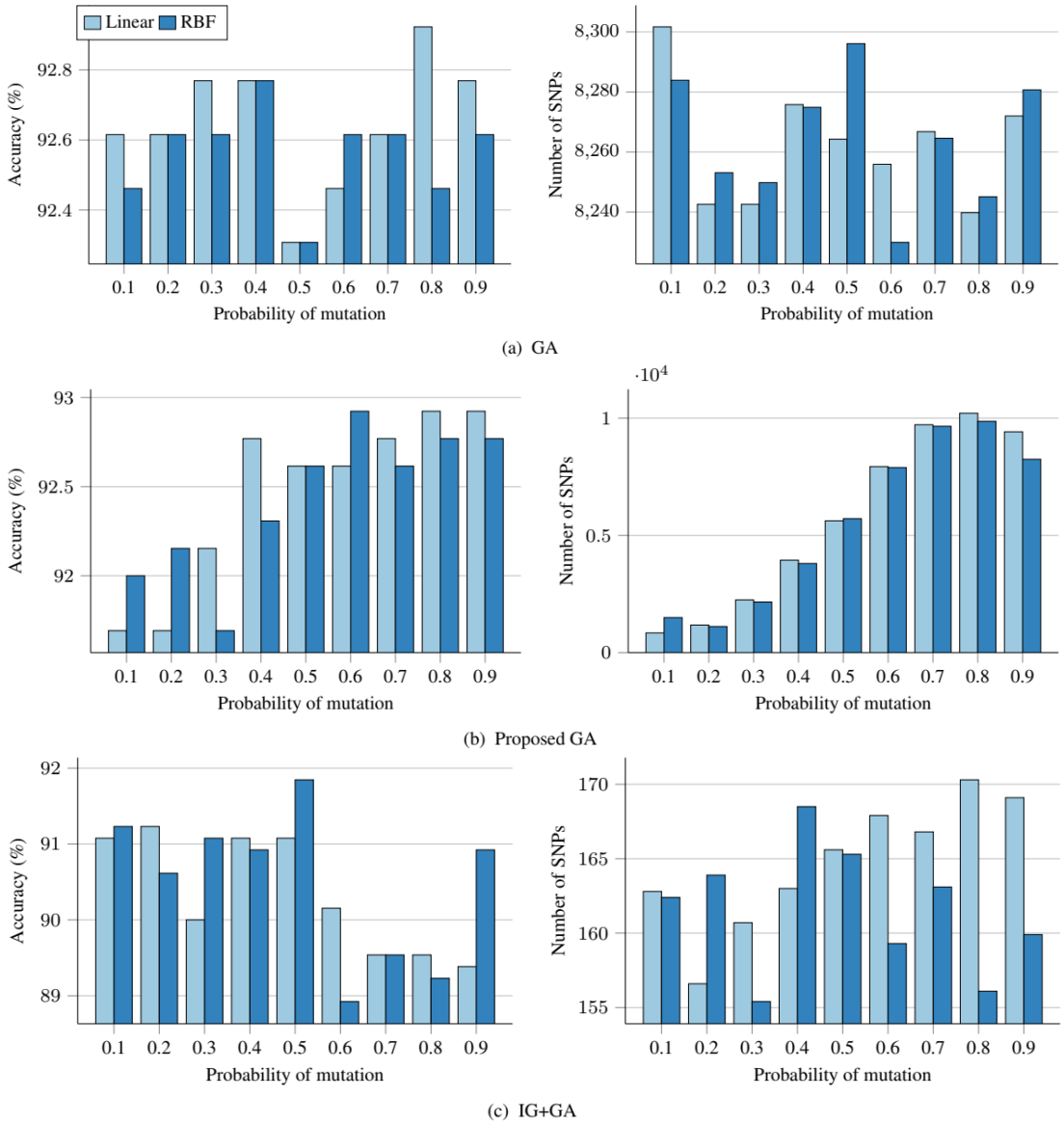
As mentioned previously, using features selected from either linear kernel or RBF kernel alone did not result in good classification accuracy from IG+GA+FFS and IG+Proposed GA+FFS, as shown in Fig. 6. IG+GA+FFS and IG+Proposed GA+FFS could achieve 59.23 % and 92.62 % accuracy with a set of features selected based on linear kernel, respectively, and at 62.54 % and 92.46 % accuracy based on RBF kernel, respectively. Thus, we used the unique features gleaned from the features selected by both kernels, which resulted in much better accuracy (76.62 % and 94.62

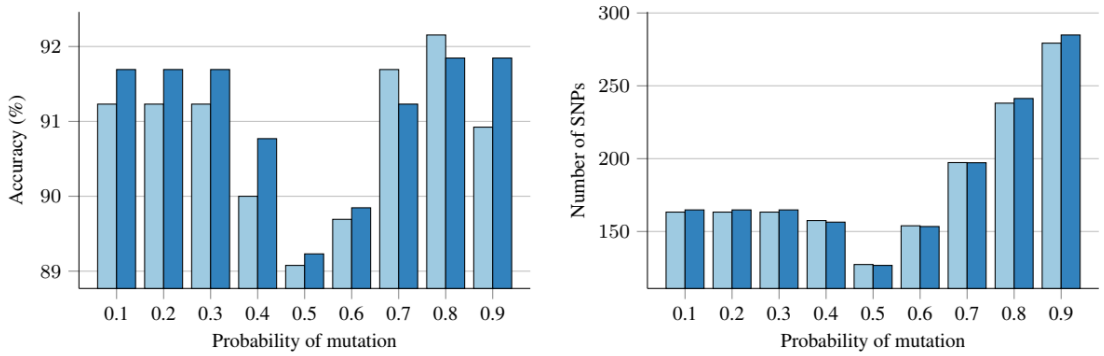
% for using IG+GA+FFS and IG+Proposed GA+FFS for using linear kernel, respectively, and 77.08 %, 94.80 % for using RBF kernel, respectively). It is clear that combining more relevant features led to better performance.

5.2 Results from analysis of variance (ANOVA)

One-Way ANOVA was used to test the hypotheses in terms of whether or not the average classification accuracies from different methods used were statistically different. In general, one-way ANOVA is used to compare more than two means and whether or not at least a pair of means is different. If so, a multiple comparison test will be used to find which pairs are significantly different.

As for the results of a one-way ANOVA analysis for the classification accuracies achieved by all methods, it was found





(d) IG+Proposed GA

Fig. 5: Classification accuracies and numbers of selected SNPs obtained from using a range of P_m values

that at least one pair of feature selection methods gave significantly different accuracies at $p \leq 0.05$. Thus, multiple comparison was performed. The results from the multiple comparison show that the classification accuracies achieved by IG+GA+FFS with both linear and RBF kernels were significantly worse than those achieved by all the other methods. In addition, the accuracy achieved by IG+Proposed GA+FFS with linear kernel was statistically and significantly better than those achieved by IG+GA. As seen in Table 3, IG+Proposed GA+FFS was able to reduce the number of selected features to 0.86 %, though these differences were slight (competitive to the others),

Table 2: Average classification accuracy and selected SNPs achieved by each method of the proposed approach (the best values are in bold). It is noted that the numbers of SNPs used in methods with FFS are constants

Method	**Accuracy (%)**		**#SNP**	
	Linear	**RBF**	**Linear**	**RBF**
Entire SNPs	92.46 ± 1.98	92.46 ± 1.98	16,579.00 (100 %)	16,579.00 (100 %)
GA	92.92 ± 2.08	92.62 ± 2.03	8,239.80 ± 73.84 (49.70 %)	8,245.10 ± 56.99 (49.73 %)
Proposed GA	91.69 ± 3.18	92.15 ± 2.85	1,176.50 ± 573.61 (7.09 %)	1,113.10 ± 488.32 (6.71 %)
IG	92.15 ± 2.11	92.15 ± 2.11	329.00 ± 87.73 (1.98 %)	329.00 ± 87.73 (1.98 %)
IG+GA	90.31 ± 2.81	91.85 ± 2.18	165.60 ± 45.47 (1.00 %)	165.30 ± 5.12 (1.00 %)
IG+Proposed GA	92.15 ± 2.76	91.85 ± 2.91	238.10 ± 6.71 (1.44 %)	241.30 ± 9.52 (1.46 %)
IG+FFS	94.15 ± 2.27	94.00 ± 2.34	202.00 (1.22 %)	202.00 (1.22 %)
IG+GA+FFS	76.62 ± 4.49	77.08 ± 4.61	**21.00 (0.13 %)**	**21.00 (0.13 %)**
IG+Proposed GA+FFS	**94.62 ± 2.21**	**94.80 ± 2.08**	142.00 (0.86 %)	142.00 (0.86 %)

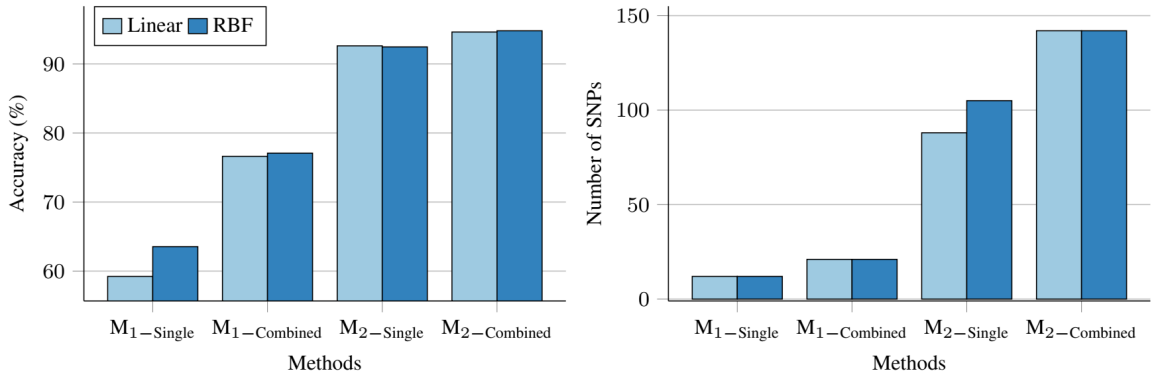


Fig. 6: Classification accuracy and number of SNPs after being processed by FFS with single set of features and combined set of features (M_1 is IG+GA+FFS; M_2 is IG+Proposed GA+FFS)

5.3 Results from PCA analysis

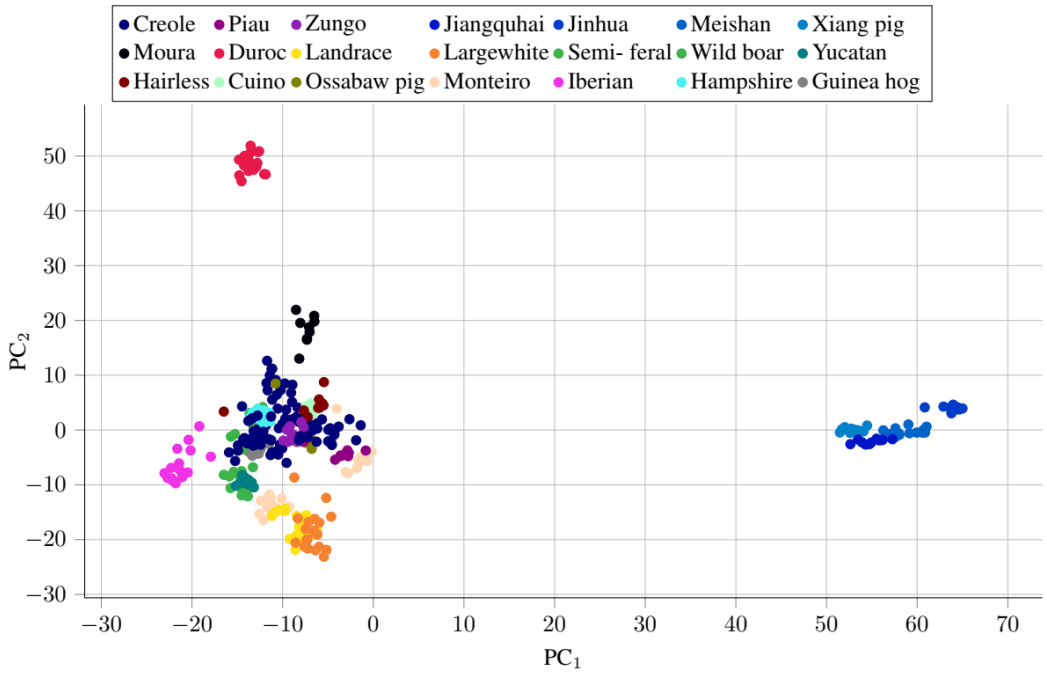
After 142 of the most significant SNPs were selected, they were used to perform an analysis of the relationship between swine breeds by PCA. In general, principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values for linearly uncorrelated variables, called principal components (PC). This transformation is defined in such a way that the first principal component (PC_1) has the largest possible variance. We performed PCA on both the entire SNP dataset and on the set of selected SNPs from our approach and compared the results.

Table 3: Results of pairwise comparison among all the methods from the multiple comparison analysis (the significantly different accuracies are in bold)

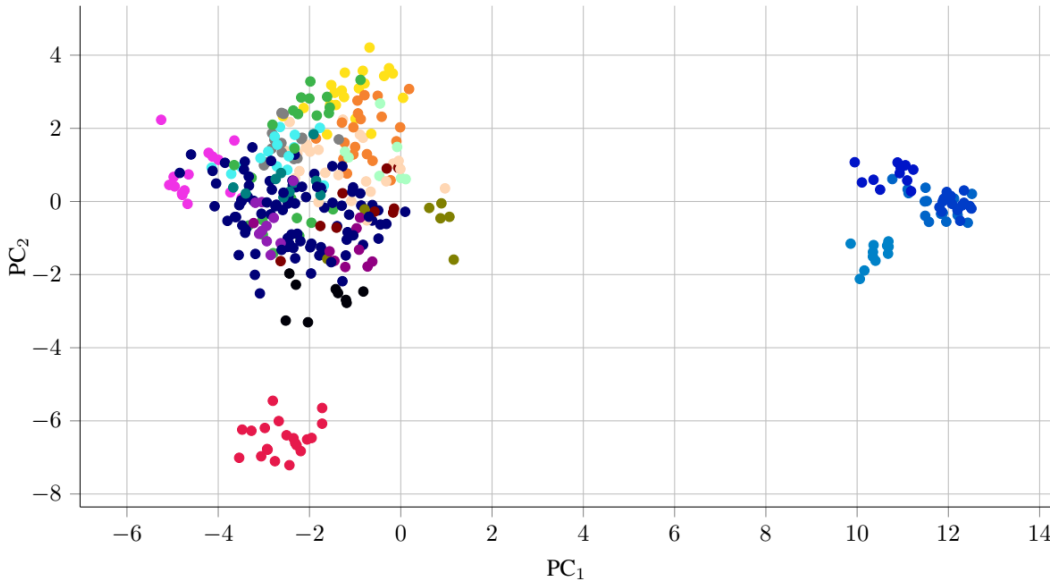
Paired Method 1	Paired Method 2	Linear kernel				RBF kernel			
		Mean difference	*p*-value	95% confidence interval for the mean difference		Mean difference	*p*-value	95% confidence interval for the mean difference	
				Lower Bound	Upper Bound			Lower Bound	Upper Bound
Entire SNPs	GA	-0.46	1.00	-4.45	3.52	-0.15	1.00	-3.98	3.67
Entire SNPs	Proposed GA	0.77	1.00	-3.21	4.75	0.31	1.00	-3.52	4.13
Entire SNPs	IG	0.31	1.00	-3.68	4.29	0.31	1.00	-3.52	4.13
Entire SNPs	IG+GA	2.15	0.73	-1.83	6.14	0.62	1.00	-3.21	4.44
Entire SNPs	IG+Proposed GA	0.31	1.00	-3.68	4.29	0.62	1.00	-3.21	4.44
Entire SNPs	IG+FFS	-1.69	0.91	-5.68	2.29	-1.54	0.93	-5.37	2.29
Entire SNPs	IG+GA+FFS	15.85	**0.00**	11.86	19.83	15.38	**0.00**	11.56	19.21
Entire SNPs	IG+Proposed GA+FFS	-2.15	0.73	-6.14	1.83	-2.31	0.60	-6.13	1.52
GA	Proposed GA	1.23	0.99	-2.75	5.21	0.46	1.00	-3.37	4.29
GA	IG	0.77	1.00	-3.21	4.75	0.46	1.00	-3.37	4.29
GA	IG+GA	2.62	0.49	-1.37	6.60	0.77	1.00	-3.06	4.60
GA	IG+Proposed GA	0.77	1.00	-3.21	4.75	0.77	1.00	-3.06	4.60
GA	IG+FFS	-1.23	0.99	-5.21	2.75	-1.38	0.96	-5.21	2.44
GA	IG+GA+FFS	16.31	**0.00**	12.32	20.29	15.54	**0.00**	11.71	19.37
GA	IG+Proposed GA+FFS	-1.69	0.91	-5.68	2.29	-2.15	0.69	-5.98	1.67
Proposed GA	IG	-0.46	1.00	-4.45	3.52	0.00	1.00	-3.83	3.83
Proposed GA	IG+GA	1.38	0.97	-2.60	5.37	0.31	1.00	-3.52	4.13
Proposed GA	IG+Proposed GA	-0.46	1.00	-4.45	3.52	0.31	1.00	-3.52	4.13
Proposed GA	IG+FFS	-2.46	0.57	-6.45	1.52	-1.85	0.83	-5.67	1.98
Proposed GA	IG+GA+FFS	15.08	**0.00**	11.09	19.06	15.08	**0.00**	11.25	18.90
Proposed GA	IG+Proposed GA+FFS	-2.92	0.33	-6.91	1.06	-2.62	0.43	-6.44	1.21
IG	IG+GA	1.85	0.86	-2.14	5.83	0.31	1.00	-3.52	4.13
IG	IG+Proposed GA	0.00	1.00	-3.98	3.98	0.31	1.00	-3.52	4.13
IG	IG+FFS	-2.00	0.80	-5.98	1.98	-1.85	0.83	-5.67	1.98
IG	IG+GA+FFS	15.54	**0.00**	11.55	19.52	15.08	**0.00**	11.25	18.90
IG	IG+Proposed GA+FFS	-2.46	0.57	-6.45	1.52	-2.62	0.43	-6.44	1.21
IG+GA	IG+Proposed GA	-1.85	0.86	-5.83	2.14	0.00	1.00	-3.83	3.83
IG+GA	IG+FFS	-3.85	0.07	-7.83	0.14	-2.15	0.69	-5.98	1.67
IG+GA	IG+GA+FFS	13.69	**0.00**	9.71	17.68	14.77	**0.00**	10.94	18.60

IG+GA	IG+Proposed GA+FFS	-4.31	0.02	-8.29	-0.32	-2.92	0.28	-6.75	0.90
IG+Proposed GA	IG+FFS	-2.00	0.80	-5.98	1.98	-2.15	0.69	-5.98	1.67
IG+Proposed GA	IG+GA+FFS	15.54	**0.00**	11.55	19.52	14.77	**0.00**	10.94	18.60
IG+Proposed GA	IG+Proposed GA+FFS	-2.46	0.57	-6.45	1.52	-2.92	0.28	-6.75	0.90
IG+FFS	IG+GA+FFS	17.54	**0.00**	13.55	21.52	16.92	**0.00**	13.10	20.75
IG+FFS	IG+Proposed GA+FFS	-0.46	1.00	-4.45	3.52	-0.77	1.00	-4.60	3.06
IG+GA+FFS	IG+Proposed GA+FFS	-18.00	**0.00**	-21.98	-14.02	-17.69	**0.00**	-21.52	-13.87

As is generally known, PC relates to the variance of data points in a dataset. PC_1 is the most significant PC and PC_2 is the second–most significant PC. Calculated from all SNPs in the dataset, when PC_1 was plotted versus PC_2 as in Fig. 7a, it can be seen that the data points representing each population of swine breed are closely grouped together, while those representing different populations of swine breeds are clearly separated: Chinese pigs (Blue), landrace (Yellow), LargeWhite (Orange), Moura (Black), and Duroc (Red). These results are quite similar to those from the PCA analysis reported in [1], which used 206 village pig samples from the American continent, including those from Canary Islands and Iberian Peninsula and 183 outgroup pigs from Iberian Peninsula, China, and some other global locations. Most samples were from the Iberian Peninsula. The total number of samples was 389, while the total number of SNPs was 46,259. The conclusion was made that most European village pigs were genetically similar; Chinese pigs–Jiangquhai, Jinhua, Meishan, and Xiang pigs–were distinctly dissimilar to breeds from other global locations, while Landrace and LargeWhite were also genetically dissimilar to breeds from other global locations, but more similar to Asian pigs than to wild boars and Iberian. Lastly, Duroc was also genetically and distinctly different from other breeds. In this study, the dataset that we used had a smaller number of samples (356), as mentioned in Section 3.0, and the number of selected SNPs used in the analysis was also much smaller (142). However, the PCA analysis results achieved by the proposed approach, as shown in Fig. 7b, are still nearly the same as the PCA results achieved using the entire SNPs in the dataset as well as the PCA results from [1]: namely, the data points for Landrace, LargeWhite, and Moura were clearly separate from the data points for other breeds. Most distinctly separate from those of other breeds were data points for Duroc and Chinese pigs. These results demonstrate that the proposed approach is valid while providing a much higher computational efficiency than using the entire SNPs from the dataset.



(a) Entire SNPs



(b) 142 SNPs

Fig. 7: Conventional PCA projection of SNPs in the dataset

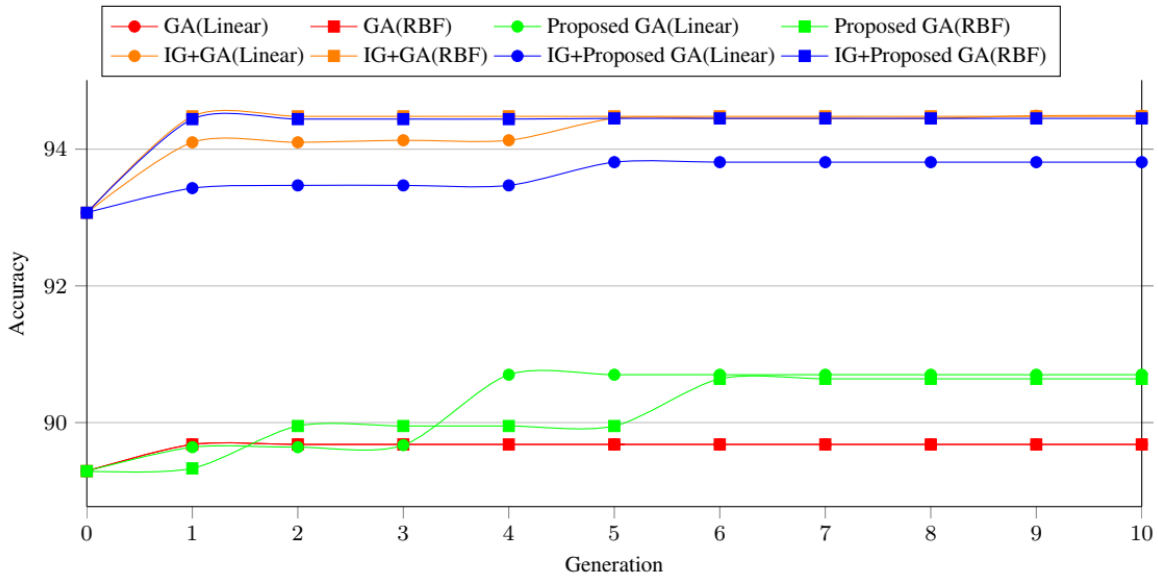


Fig. 8: The classification accuracy from each generation of the first randomly-seeded dataset

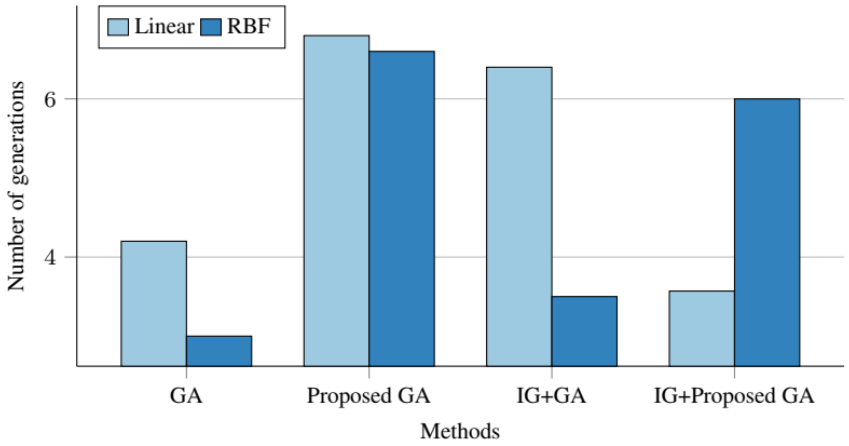


Fig. 9: The number of generations at the stop of runs of all 10 randomly-seeded datasets of each tested method

6.0 CONCLUSION

A small number of usable Porcine SNPs for swine classification can be suitably selected by using feature selection and classification techniques. This study employed the IG, GA, Proposed GA, IG+GA, IG+Proposed GA, IG+FFS, IG+GA+FFS, and IG+Proposed GA+FFS methods to find a small number of suitable SNPs and SVM for classification. It was found that IG+Proposed GA+FFS was able to reduce the number of suitable SNPs to 0.86 % of the total number of SNPs in the dataset used while providing a high classification accuracy of 94.80 %, which was higher than those achieved by other methods. Compared to classification results reported in pervious literature, the results from the proposed approach were comparable, demonstrating the validity of the approach that also provides a much higher computational efficiency. Future work should determine the genes that are related to these selected SNPs, including finding their biological pathway and determining the gene ontology annotation that relates to the genes. The information gained from such future work would be very useful in the biology field.

REFERENCES

[1] W. Burgos-Paz, C. A. Souza, H. J. Megens, Y. Ramayo-Caldas, M. Melo, C. Lemus-Flores, E. Caal, H. W. Soto, R. Mart´ınez, L. A. Álvarez, L. Aguirre, V. Iñiguez, M. A. Revidatti, O. R. Martínez-López, S. Llambi, A. Esteve-Codina, M. C. Rodríguez, R. P. M. A. Crooijmans, S. R. Paiva, L. B. Schook, M. a. M. Groenen, and M. Pérez-Enciso, "Porcine colonization of the Americas: A 60k SNP story," *Heredity*, vol. 110, no. 4, pp. 321–330, 2013.

[2] P. Larrañaga, B. Calvo, R. Santana, C. Bielza, J. Galdiano, I. Inza, J. A. Lozano, R. Armañanzas, G. Santafé, A. Pérez, and V. Robles, "Machine learning in bioinformatics," *Briefings in Bioinformatics*, vol. 7, no. 1, pp. 86–112, 2006.

[3] Y. Saeys, I. Inza, and P. Larrañaga, "A review of feature selection techniques in bioinformatics," *Bioinformatics*, vol. 23, no. 19, pp. 2507–2517, 2007.

[4] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[5] H. Zang, S. Zhang, and K. Hapeshi, "A review of nature-inspired algorithms," *Journal of Bionic Engineering*, vol. 7, no. Supplement, pp. S232–S237, 2010.

[6] L.-Y. Chuang, H.-W. Chang, C.-J. Tu, and C.-H. Yang, "Improved binary PSO for feature selection using gene expression data," *Computational Biology and Chemistry*, vol. 32, no. 1, pp. 29–38, 2008.

[7] C.-L. Huang, "ACO-based hybrid classification system with feature subset selection and model parameters optimization," *Neurocomputing*, vol. 73, no. 1, pp. 438–448, 2009.

[8] R. Y. M. Nakamura, L. A. M. Pereira, D. Rodrigues, K. A. P. Costa, J. P. Papa, and X.-S. Yang, "Binary bat algorithm for feature selection," in *Swarm Intelligence and Bio-Inspired Computation*, 2013, pp. 225–237.

[9] D. Rodrigues, L. A. M. Pereira, T. N. S. Almeida, J. P. Papa, A. N. Souza, C. C. O. Ramos, and X. S. Yang, "BCS: A binary cuckoo search algorithm for feature selection," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS 2013)*, 2013, pp. 465–468.

[10] D. Rodrigues, X.-S. Yang, A. N. de Souza, and J. P. Papa, "Binary flower pollination algorithm and its application to feature selection," in *Recent Advances in Swarm Intelligence and Evolutionary Computation*, ser. Studies in Computational Intelligence, 2015, pp. 85–100.

[11] M. L. Raymer, W. F. Punch, E. D. Goodman, L. A. Kuhn, and A. K. Jain, "Dimensionality reduction using genetic algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 2, pp. 164–171, 2000.

[12] L. Rokach, "Genetic algorithm-based feature set partitioning for classification problems," *Pattern Recognition*, vol. 41, no. 5, pp. 1676–1700, 2008.

[13] L. F. Leijôto, T. A. D. O. Rodrigues, L. E. Záratey, and C. N. Nobre, "A genetic algorithm for the selection ´ of features used in the prediction of protein function," in *Proceedings of the IEEE International Conference on Bioinformatics and Bioengineering (BIBE 2014)*, 2014, pp. 168–174.

[14] G. Mahdevar, J. Zahiri, M. Sadeghi, A. Nowzari-Dalini, and H. Ahrabian, "Tag SNP selection via a genetic algorithm," *Journal of Biomedical Informatics*, vol. 43, no. 5, pp. 800–804, 2010.

[15] S. Peng, Q. Xu, X. B. Ling, X. Peng, W. Du, and L. Chen, "Molecular classification of cancer types from microarray data using the combination of genetic algorithms and support vector machines," *FEBS Letters*, vol. 555, no. 2, pp. 358–362, 2003.

[16] L. Li, W. Jiang, X. Li, K. L. Moser, Z. Guo, L. Du, Q. Wang, E. J. Topol, Q. Wang, and S. Rao, "A robust hybrid between genetic algorithm and support vector machine for extracting an optimal feature gene subset," *Genomics*, vol. 85, no. 1, pp. 16–23, 2005.

[17] J. Huang, Y. Cai, and X. Xu, "A hybrid genetic algorithm for feature selection wrapper based on mutual information," *Pattern Recognition Letters*, vol. 28, no. 13, pp. 1825–1844, 2007.

[18] S. Lei, "A feature selection method based on information gain and genetic algorithm," in *Proceedings of the International Conference on Computer Science and Electronics Engineering (ICCSEE 2012)*, vol. 2, 2012, pp. 355–358.

[19] C. Lazar, J. Taminau, S. Meganck, D. Steenhoff, A. Coletta, C. Molter, V. de Schaetzen, R. Duque, H. Bersini, and A. Nowe, "A survey on filter techniques for feature selection in gene expression microarray analysis," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 9, no. 4, pp. 1106–1119, 2012.

[20] S.-B. Cho and H.-H. Won, "Machine learning in DNA microarray analysis for cancer classification," in *Proceedings of the 1st Asia-Pacific Bioinformatics Conference on Bioinformatics (APBC 2003)*, vol. 19, 2003, pp. 189–198.

[21] T. Jirapech-Umpai and S. Aitken, "Feature selection and classification for microarray data analysis: Evolutionary methods for identifying predictive genes," *BMC Bioinformatics*, vol. 6, p. 148, 2005.

[22] L. C. Borro, S. R. M. Oliveira, M. E. B. Yamagishi, A. L. Mancini, J. G. Jardine, I. Mazoni, E. H. dos Santos, R. H. Higa, P. R. Kuser, and G. Neshich, "Predicting enzyme class from protein structure using bayesian classification," *Genetics and Molecular Research*, vol. 5, no. 1, pp. 193–202, 2006.

[23] İ. İlhan and G. Tezel, "A genetic algorithm–support vector machine method with parameter optimization for selecting the tag SNPs," *Journal of Biomedical Informatics*, vol. 46, no. 2, pp. 328–340, 2013.