International Journal of Computer Information Systems and Industrial Management Applications.
ISSN 2150-7988 Volume 4 (2012) pp. 467-476

A Comparative Study of Feature Selection Methods for Authorship Invarianceness in Writer Identification

Satrya Fajri Pratama, Azah Kamilah Muda, Yun-Huoy Choo and Noor Azilah Muda

Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka,
Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia
satrya.fajri@yahoo.com, {azah, huoy, azilah}@utem.edu.my

***Abstract*: Handwriting is individualistic. The uniqueness of shape and style of handwriting can be used to identify the significant features in authenticating the author of writing. Acquiring these significant features leads to an important research in Writer Identification domain. This paper is meant to explore the usage of feature selection in Writer Identification. Various filter and wrapper feature selection methods are selected and their performances are analyzed. This paper describes an improved sequential forward feature selection method besides the exploration of significant features for invarianceness of authorship from global shape features by using various feature selection methods. The promising results show that the proposed method is worth to receive further exploration in identifying the handwritten authorship.**

***Keywords*:** feature selection, authorship invarianceness, significant features, writer identification, comparative study**.**

I. Introduction

Feature selection has become the focus of research area for a long time. The purpose of feature selection is to obtain the most minimal sized subset of features [1]. Practical experience has shown that if there is too much irrelevant and redundant information present, the performance of a classifier might be degraded. Removing these irrelevant and redundant features can improve the classification accuracy.

The three popular methods of feature selection are filter method, wrapper method, and embedded method has been presented in [2]. Filter method assesses the relevance of features [3], wrapper method uses an induction algorithm [4], while embedded method do the selection process inside the induction algorithm [5].

Studies have shown that there are no feature selection methods more superior compared to others [6]. The selection of the methods to use sometimes depends on the size of the data itself. Using filter methods means to have a good computational complexity, but the higher complexity of the wrapper methods will also produce higher accuracy in the final result, whereas embedded methods are intrinsic to some learning algorithm and so only those algorithm designed with this characteristic can be used.

Writer Identification (WI) can be included as a particular kind of dynamic biometric in pattern recognition for forensic application. WI distinguishes writers based on the shape or individual style of writing while ignoring the meaning of the word or character written. The shape and style of writing are different from one person to another. Even for one person, they are different in times. However, everyone has their own style of writing and it is individualistic. It must be unique feature that can be generalized as significant individual features through the handwriting shape.

The main issue in WI is how to acquire the features that reflect the author of handwriting. Thus, it is an open question whether the extracted features are optimal or near-optimal to identify the author. Extracted features may include many garbage features. Such features are not only useless in classification, but sometimes degrade the performance of a classifier designed on a basis of a finite number of training samples [7]. The features may not be independent of each other or even redundant. Moreover, there may be features that do not provide any useful information for the task of writer identification [8]. Therefore, selection of the significant features is very important in order to identify the writer, moreover to improve the classification accuracy.

Thus, this paper focuses on identifying the significant features of word shape by using various filter and wrapper feature selection methods, including the proposed feature selection method prior the identification task on some small-sized data sets, where the number of features is between 1-19 features [7]. The remainder of the paper is structured as follows. In next section, an overview of authorship invarianceness is given. Section III provides an overview of feature selection methods, including the proposed feature selection method. In Section IV, experimental setup describing the dataset, experimental design and environmental setup are presented and the results are discussed in Section V. Finally, conclusion and future work is drawn in Section VI.

II. Authorship Invarianceness

Handwriting is individual to personal. Handwriting has long been considered individualistic and writer individuality rests on the hypothesis that each individual has consistent handwriting [9], [10], [11], [12], [13]. The relation of character, shape and the style of writing are different from one to another.

Handwriting analysis consists of two categories, which are handwriting recognition and handwriting identification. Handwriting recognition deals with the contents conveyed by the handwritten word, while handwriting identification tries to differentiate handwritings to determine the author. There are two tasks in identifying the writer of handwriting, namely identification and verification. Identification task determines the writer of handwriting from many known writers, while verification task determines whether one document and another is written by the same writer.

WI has attracted many researchers to work in, primarily in forensic and biometric applications. The challenge is to find the best solution to identify the writer accurately; therefore the main issue is how to acquire the individual features from various handwritings, and thus various studies have been conducted and discussed in [14].

The challenge in WI is how to acquire the features that reflect the author for these variety styles of handwriting [15], [16], [17], [18], [19], [12], either for one writer or many writers. These features are required to classify in order to identify the variance between features for same writer is lower than different writer which known as Authorship Invarianceness. Among these features are exists the significant individual features which directly unique to those individual.

There are three steps involved in traditional handwriting identification task, which are pre-processing, feature extraction and classification [20]. Previous studies have explored various methods to enhance traditional task, and improves the classification accuracy. One study has been conducted [21] by incorporating feature selection task after feature extraction task, and the results shows significantly improved classification accuracy.

III. Feature Selection

Feature selection has become an active research area for decades, and has been proven in both theory and practice [22]. The main objective of feature selection is to select the minimally sized subset of features as long as the classification accuracy does not significantly decreased and the result of the selected features class distribution is as close as possible to original class distribution [1].

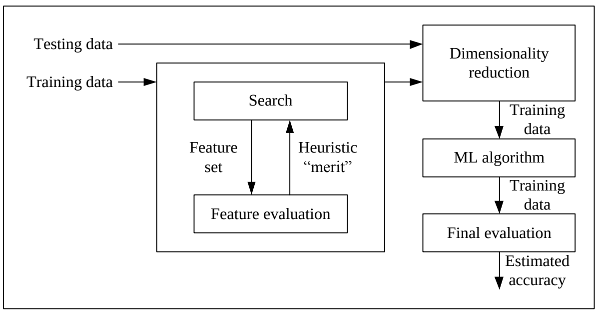
In contrast to other dimensionality reduction methods like those based on projection or compression, feature selection methods do not alter the original representation of the variables, but merely select a subset of them. Thus, they preserve the original semantics of the variables. However, the advantages of feature selection methods come at a certain price, as the search for a subset of relevant features introduces an additional layer of complexity in the modeling task [2]. In this work, feature selection is explored in order to find the most significant features which by is the unique features of individual's writing. The unique features a mainly contribute to the concept of Authorship Invarianceness in WI.

There are three general methods of feature selection which are filter method, wrapper method, and embedded method [23]. Filter method assesses the relevance of features by looking only at the intrinsic properties of the data. A feature relevance score is calculated, and low-scoring features are removed [3]. Simultaneously, wrapper method uses an induction algorithm to estimate the merit of feature subsets. It explores the space of features subsets to optimize the induction algorithm that uses the subset for classification [4]. On the other hand, in embedded method, the selection process is done inside the induction algorithm itself, being far less computationally intensive compared with wrapper methods [5]. For the purpose of this paper however, only filter and wrapper method will be further explored.

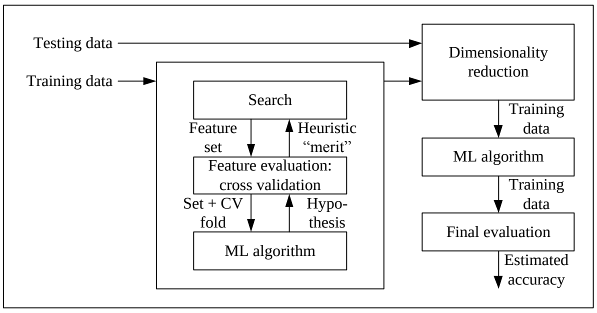
The advantages of filter method are (a) fast execution: filter method generally involve a non-iterative computation on the dataset, which can execute much faster than a classifier training session, and (b) generality: since filters evaluate the intrinsic properties of the data, rather than their interactions with a particular classifier, their results exhibit more generality: the solution will be "good" for a larger family of classifiers. However, the disadvantage is the tendency to select large subsets; this is because since the filter objective functions are generally monotonic, filter method tends to select the full feature set as the optimal solution. This forces the user to select an arbitrary cutoff on the number of features to be selected.

On the other hand, the advantages of wrapper method are (a) accuracy: wrappers generally achieve better recognition rates than filter method since they are tuned to the specific interactions between the classifier and the dataset, and (b) ability to generalize: wrappers have a mechanism to avoid overfitting, since they typically use cross-validation measures of predictive accuracy. But these advantages come with several disadvantages, namely (a) slow execution: since the wrapper must train a classifier for each feature subset (or several classifiers if cross-validation is used), the method can become unfeasible for computationally intensive methods, and (b) lack of generality: the solution lacks generality since it is tied to the bias of the classifier used in the evaluation function. The "optimal" feature subset will be specific to the classifier under consideration.

There are many available filter and wrapper methods, however only three filter methods and two wrapper methods will be discussed here. These methods are Correlation-based Feature Selection (CFS) [3], Consistency-based Feature Selection, also known as Las Vegas Filter (LVF) [24], and Fast Correlation-based Filter (FCBF) [25] for filter methods. As for wrapper, the selected methods are Sequential Forward Selection (SFS), Sequential Forward Floating Selection (SFFS), and Computationally Inexpensive Sequential Forward Floating Selection (CI-SFFS), which are the proposed method. Figure 1(a) depicts filter method, while Figure 1(b) depicts wrapper method.



(a) Filter feature selection



(b) Wrapper feature selection

Figure 1. Feature selection model

A. CFS

CFS ranks feature subsets according to a correlation based heuristic evaluation function [2]. CFS selects subsets that contain highly correlated features with the class and uncorrelated with each other. The acceptance of a feature will depend on the extent to which it predicts classes in areas of the instance space not already predicted by other features. The feature subset evaluation function is

$$M_s = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}} \quad (1)$$

where M_s is the heuristic "merit" of a feature subset S containing k features, $\overline{r_{cf}}$ and $\overline{r_{ff}}$ is the mean feature-class correlation ($f \in S$) and the average feature-feature inter-correlation respectively.

CFS calculates the correlations and then searches the feature subset space. The subset with the highest merit found is used to reduce the dimensionality.

B. LVF

LVF uses a random generation of subsets and an inconsistency measure as evaluation function [24]. Two instances are inconsistent if they have the same feature values but different classes. The inconsistency measure of a given subset of features T relative to a dataset D is defined as

$$Inconsistency(T, D) = \frac{\sum_{i=1}^{K} |D_i| - |M_i|}{N} \quad (2)$$

where $|D_i|$ is the number of occurrences of the i-th feature value combination on T, K is the number of the distinct combinations of features values on T, $|M_i|$ is the cardinality of the class to which belong the majority of instances on the i-th feature value combination, and N is the number of instances in the dataset D. The algorithm requires an inconsistency threshold close or equal to zero. Any candidate subset having is rejected if inconsistency greater than the threshold. When the maximum number of generated subsets is reached, the generation process is stopped.

C. FCBF

FCBF uses Symmetrical Uncertainty (SU) to calculate dependences of features and finds best subset using backward selection method with sequential search strategy [25]. SU is a normalized information theoretic measure which uses entropy and conditional entropy values to calculate dependencies of features. If X is a random variable and $P(x)$ is the probability of x, the entropy of X is

$$H(X) = -\sum_i P(x_i)\log_2 P(x_i) \quad (3)$$

Conditional entropy or conditional uncertainty of X given another random variable Y is the average conditional entropy of X over Y

$$H(X \mid Y) = -\sum_j P(y_j)\sum_i P(x_i \mid y_j)\log_2 P(x_i \mid y_j) \quad (4)$$

$$SU(X,Y) = 2\left[\frac{IG(X \mid Y)}{H(X) + H(Y)}\right] \quad (5)$$

An SU value of 1 indicates that using one feature other feature's value can be totally predicted and value 0 indicates two features are totally independent. The SU values are symmetric for both features.

D. SFS

SFS initialized the best subset of features Y_0 as the empty set [26]. The feature x^+ that gives the highest correct classification rate $J(Y_k + x^+)$ is added to Y_k at the each step along with the features which already included in Y_k. The process continues until the correct classification rate given by Y_k and each of the features not yet selected does not increase. SFS performs best when the optimal subset has a small number of features. When the search is near the empty set, a large number of states can be potentially evaluated, and towards the full set, the region examined by SFS is narrower since most of the features have already been selected. The algorithm of SFS is shown as below.

1. Start with the empty set $Y_0 = \{\phi\}$
2. Select the next best feature $x^+ = \arg\max_{x^+ \notin Y_k} [J(Y_k + x^+)]$
3. If $J(Y_k + x^+) > J(Y_k)$

3.1. Update $Y_{k+1} = Y_k + x^+; k = k+1$

3.2. Go to step 2

4. End

However, this method suffers from the nesting effect. This means that a feature that is included in some step of the iterative process cannot be excluded in a later step. Thus, the results are sub-optimal.

E. SFFS

SFFS method was introduced by [27] to deal with the nesting problem. In SFFS, Y_0 is initialized as the empty set and in each step a new subset is generated first by adding a feature x^+, but after that features x^- is searched for to be eliminated from Y_k until the correct classification rate $J(Y_k - x^-)$ decreases. The iterations continue until no new variable can be added because the recognition rate $J(Y_k + x^+)$ does not increase. The algorithm is as below.

1. Start with the empty set $Y_0 = \{\phi\}$
2. Select the next best feature $x^+ = \arg\max_{x^+ \notin Y_k} [J(Y_k + x^+)]$
3. If $J(Y_k + x^+) > J(Y_k)$

3.1. Update $Y_{k+1} = Y_k + x^+; k = k+1$

```
3.2. Remove the worst feature
```
$$x^- = \arg\max_{x^- \in Y_k} [J(Y_k - x^-)]$$
```
3.3. If
```
$J(Y_k - x^-) > J(Y_k)$
```
3.3.1. Update
```
$Y_{k+1} = Y_k - x^-; k = k+1$
```
3.3.2. Go to 3.2
3.4. Else
3.4.1. Go to 2
4. End
```

F. CI-SFFS

As mentioned earlier, most of wrapper methods are constrained by the time complexity, and as the result, its usage is getting less frequent compared to filter method. Thus, an improved wrapper method should be devised to allow faster execution time. CI-SFFS is introduced as the improvement to SFFS to cater with the slow execution time. The concept of CI-SFFS is similar with traditional SFFS, however it is implemented and enhanced by recent programming techniques, such as memory pooling and multithreading.

The process of searching for the best feature x^+ within SFFS is repetitive, thus making its results are constants, regardless the number of execution. Therefore, it is only efficient if these results are stored in the memory, rather than having to repeat the process and recalculate every result. By storing these results, CI-SFFS only have to determine whether a feature ($x^+ \notin Y_k$) has been previously calculated. If it hasn't been calculated, then the result will be calculated and stored. This process is referred as memory pooling.

In computer science, a thread is the smallest unit of processing that can be scheduled by an operating system. Multithreading allows multiple threads to exist within the context of a single process [28]. These threads share the process' resources but are able to execute independently. Multithreading programming benefits are as follow:

1) Improving application responsiveness

Any program in which many activities are not dependent upon each other can be redesigned so that each independent activity is defined as a thread.

2) Using multi-processors efficiently

Applications that express concurrency requirements with threads need not take into account the number of available processors. The performance of the application improves transparently with additional processors because the operating system takes care of scheduling threads for the number of processors that are available. When multicore processors and multithreaded processors are available, a multithreaded application's performance scales appropriately because the cores and threads are viewed by the operating system as processors. Numerical algorithms and numerical applications with a high degree of parallelism, such as matrix multiplications, can run much faster when implemented with threads on a multiprocessor.

3) Improving program structure

Many programs are more efficiently structured as multiple independent or semi-independent units of execution instead of as a single, monolithic thread. Multithreaded programs, especially programs that provide service to multiple concurrent users, can be more adaptive to variations in user demands than single-threaded programs.

4) Using fewer system resources

Each process has a full address space and operating environment state. Cost of creating and maintaining this large amount of state information makes each process much more expensive than a thread in both time and space. In addition, the inherent separation between processes can require a major effort by the programmer. This effort includes handling communication between the threads in different processes, or synchronizing their actions. When the threads are in the same process, communication and synchronization becomes much easier.

By implementing these recent techniques, the execution time is greatly reduced (more than 53 times faster) compared to the original method without having to sacrifice the classification accuracy. Although time complexity is not an issue in WI domain, faster execution time allows further enhancement to this method, for instance by hybridizing it with recent optimization techniques, which may increase the classification accuracy. The algorithm is as shown below.

```
1. Start with the empty set
```
$Y_0 = \{\phi\}$
```
2. Calculate the merit of each feature
3. Store the merits in the memory pool
4. Spawn threads of forward feature selector
4.1. Select the next best feature
```
$$x^+ = \arg\max_{x^+ \notin Y_k} [J(Y_k + x^+)]$$
```
4.2. If
```
$J(Y_k + x^+) > J(Y_k)$
```
4.2.1. Update
```
$Y_{k+1} = Y_k + x^+; k = k+1$
```
4.2.2. Spawn threads of backward feature
       selector
4.2.2.1. Remove the worst feature
```
$$x^- = \arg\max_{x^- \in Y_k} [J(Y_k - x^-)]$$
```
4.2.2.2. If
```
$J(Y_k - x^-) > J(Y_k)$
```
4.2.2.2.1. Update
```
$Y_{k+1} = Y_k - x^-; k = k+1$
```
4.2.2.2.2. Go to 4.2.2
4.2.2.3. Else
4.2.2.3.1. Go to 4
4.3. Else
4.3.1. Go to 5
5. End
```

IV. Experiments

With the goal stated in the section above, an extensive and rigorous empirical comparative study is designed and conducted. In this section, a detailed description of the experimental method is provided.

A. Dataset

In pattern recognition problem, there are many shape representations or description techniques have been explored in order to extract the features from the image. Generally it can be classified into two different approaches when dealing with handwritten word problem, which are analytic (local / structural approach) and holistic (global approach) [29], [30]. For the each approach, it is divided into two method, which are region-based (whole region shape) methods and contour-based (contour only) methods. Holistic approach represent shape as a whole, meanwhile analytic approach represents image in sections. In this work, holistic approach of United Moment Invariant (UMI) is chosen due to the requirement of cursive word is needed to extract as one single indivisible entity. This moment function of UMI is applied in feature extraction task.

The choice of using holistic approach is not only based on the holistic advantages, but also due to its capability of using word in showing the individuality for writer identification problem as mentioned in [10] and holds immense promise for realizing near-human performance [31]. The holistic features and matching schemes must be coarse enough to be stable across exemplars of the same class such as a variety of writing styles [32]. This is aligning with this work where to extract the unique global features from word shape in order to identify the writer.

Global features extracted with this holistic approach are invariant with respect to all different writing styles [32]. Words in general may be cursive, minor touching discrete, purely discrete, one or two characters are isolated and others are discrete or mixture of these style and it still as one word. Global technique in holistic approach will extract all of these styles for one word as one whole shape. Shape is an important representation of visual image of an object. It is a very powerful feature when it is used in similarity search. Unlike color and texture features, the shape of an object is strongly tied to the object functionality and identity [33]. Furthermore, the use of holistic approach is shown to be very effective in lexicon reduction [34], moreover to increase the accuracy of classification.

Moment Function has been used in diverse fields ranging from mechanics and statistics to pattern recognition and image understanding [35]. The use of moments in image analysis and pattern recognition was inspired by [36] and [37]. [36] first presented a set of seven-tuplet moments that invariant to position, size, and orientation of the image shape. However, there are many research have been done to prove that there were some drawback in the original work by [36] in terms of invariant such as [38], [39], [40], [41], [42], and [43]. All of these researchers proposed their method of moment and tested on feature extraction phase to represents the image. A good shape descriptor should be able to find perceptually similar shape where it is usually means rotated, translated, scaled and affined transformed shapes. Furthermore, it can tolerate with human beings in comparing the image shapes. Therefore, [44] derived United Moment Invariants (UMI) based on basic scaling transformation by [36] that can be applied in all conditions with a good set of discriminate shapes features. Moreover, UMI never been tested in WI domain. With the capability of UMI as a good description of image shape, this work is explored its capability of image representation in WI domain.

One of the usages of UMI in machine learning application is handwriting recognition and handwriting identification. However, handwriting recognition deals with the contents conveyed by the image, while handwriting identification tries to differentiate each image to determine the author of those handwritings. Despite that, both of these tasks embark on the same theoretical foundation.

Table 1. Example of data used in the experiment.

Word	f1	f2	f3	f4	f5	f6	f7	f8
alone	1.84	1.79	0.91	1.31	0.84	1.00	0.73	1.79
bowed	1.53	1.08	1.12	1.96	0.72	1.49	1.82	1.46
same	1.61	1.53	0.53	0.38	0.80	1.26	0.25	3.29
scheme	1.99	8.24	0.65	0.76	3.77	0.20	0.09	2.40
the	3.08	2.06	0.52	0.64	0.52	0.82	0.31	2.75

The comparisons were carried out in dataset coming from the IAM Handwriting Database [45]. IAM Handwriting Database contains forms of handwritten English text which can be used to train and test handwritten text recognizers and to perform writer identification and verification experiments. There are 657 classes available, however only a sample of 60 classes are used for experiments. From these classes, 4400 instances are collected. Words from the forms are extracted using United Moment Invariant (UMI) which represents the word features. Feature extraction is a process of converting input object into feature vectors. The extracted features are in real value and unique for each word. Table 1 shows the examples of the data used in this experiment. However, the values shown in Table 1 are truncated to two-digit precision, while in the experiments, the precision is not truncated, which is up to sixteen-digit precision.

Extracted features can be divided into micro and macro feature classes which are local and global features. Local features denote the constituent parts of objects and the relationships, meanwhile global features describing properties of the whole object [46]. Good features are those satisfying two requirements which are small intra-class invariance and large inter-class invariance [47]. This can be defined as invarianceness of authorship in WI.

B. Experimental Design

The framework for WI follows the traditional framework of pattern recognition tasks, which are preprocessing, feature extraction, and classification. However, it has been proven that most of preprocessing tasks must be omitted because some of the original and important information are lost, and thus decrease the identification performance in WI domain [14].

Invarianceness of authorship in WI shows the similarity error for intra-class (same-writer) is small compared to inter-class (different-writers) for the same words or different words. This is due to the individual features of handwriting's style which has been proof in many researchers such as [10], [12], and [48]. Related to this paper, the objective is to make contributions towards this scientific validation using the proposed method for selecting the significant features in order to proof the authorship of invarianceness in WI.

The uniqueness of this work is to find the significant feature which actually is the unique features of individual's writing. The invarianceness of authorship relates to individuality of handwriting with the unique features of individual's writing. The highest accuracy of selected features proofs the invarianceness of authorship for intra-class is lower than inter-class where each individual's writing contains the unique styles of handwriting that is different with other individual. To achieve this, the process of selecting significant features is carried out using the proposed wrapper method before identification task.

UMI is commonly used to determine whether a shape is similar to another shape. However, in this experiment, UMI is not used to find the similar shape; instead it is used to find the similar unique features for the same class (writer). In the previous study, data discretization has been used to improve the classifier accuracy [20].

The three commonly used performance measurements for evaluating the performance of feature selection method are number of selected features, classification accuracy, and processing time. However, this paper only considers two main measures, which are number of selected features and classification accuracy.

As mentioned in the previous section, a total number of 4400 instances are used for the experiments, and are randomly divided into five different datasets to form training and testing dataset in the classification task. Every experiment has been performed using ten-fold cross-validation. The result shown is the average of the results produced by each of ten folds. In order to justify the quality of feature subset produced by each method, the feature subsets are tested against classification, which uses Modified Immune Classifier (MIC) [14] as the classifier.

C. Environmental Setup

This paper uses Waikato Environment for Knowledge Analysis (WEKA) 3.7.1 to measure the performance of each feature selection methods. WEKA came about through the perceived need for a unified workbench that would allow researchers easy access to state-of-the-art methods in machine learning [49]. WEKA includes algorithms for regression, classification, clustering, association rule mining and attribute (feature) selection.

The experiment was implemented on Intel Core 2 Duo 2.0GHz processor running on Microsoft Windows XP SP3 with 1GB of memory.

V. Experimental Results and Discussions

A. Selection Results

The number of features selected by feature selection methods is the primary consideration of this study. Feature selection methods discussed earlier will be used to determine the significant features. Table 2 shows the number of features selected by each method. All methods, except FCBF have been successfully reduced the number of features to be used. Based on the feature selection results, it is shown that these feature selection methods yield different subsets with different size.

Table 2. Experimental results on feature selection.

Method	Execution	Set A	Set B	Set C	Set D	Set E	Intersection
CFS	Execution #1	f1, f2, f3, f5, f7, f8	f1, f3, f4, f5, f6, f7	f1, f3, f4, f5, f6, f7	f1, f3, f5, f7, f8	f1, f3, f4, f5, f6, f7	**f1, f3, f5, f7**
	Execution #2	f1, f2, f3, f5, f7, f8	f1, f3, f4, f5, f6, f7	f1, f3, f4, f5, f6, f7	f1, f3, f5, f7, f8	f1, f3, f4, f5, f6, f7	**f1, f3, f5, f7**
	Execution #3	f1, f2, f3, f5, f7, f8	f1, f3, f4, f5, f6, f7	f1, f3, f4, f5, f6, f7	f1, f3, f5, f7, f8	f1, f3, f4, f5, f6, f7	**f1, f3, f5, f7**
	Execution #4	f1, f2, f3, f5, f7, f8	f1, f3, f4, f5, f6, f7	f1, f3, f4, f5, f6, f7	f1, f3, f5, f7, f8	f1, f3, f4, f5, f6, f7	**f1, f3, f5, f7**
	Execution #5	f1, f2, f3, f5, f7, f8	f1, f3, f4, f5, f6, f7	f1, f3, f4, f5, f6, f7	f1, f3, f5, f7, f8	f1, f3, f4, f5, f6, f7	**f1, f3, f5, f7**
	Intersection	**f1, f2, f3, f5, f7, f8**	**f1, f3, f4, f5, f6, f7**	**f1, f3, f4, f5, f6, f7**	**f1, f3, f5, f7, f8**	**f1, f3, f4, f5, f6, f7**	**f1, f3, f5, f7**
LVF	Execution #1	f2, f3, f4, f6	f2, f3, f4, f6	f2, f3, f4, f6	f2, f3, f4, f6	f2, f3, f4, f6	**f2, f3, f4, f6**
	Execution #2	f2, f3, f4, f6	f2, f3, f4, f6	f2, f3, f4, f6	f2, f3, f4, f6	f2, f3, f4, f6	**f2, f3, f4, f6**
	Execution #3	f2, f3, f4, f6	f2, f3, f4, f6	f2, f3, f4, f6	f2, f3, f4, f6	f2, f3, f4, f6	**f2, f3, f4, f6**
	Execution #4	f2, f3, f4, f6	f2, f3, f4, f6	f2, f3, f4, f6	f2, f3, f4, f6	f2, f3, f4, f6	**f2, f3, f4, f6**
	Execution #5	f2, f3, f4, f6	f2, f3, f4, f6	f2, f3, f4, f6	f2, f3, f4, f6	f2, f3, f4, f6	**f2, f3, f4, f6**
	Intersection	**f2, f3, f4, f6**	**f2, f3, f4, f6**	**f2, f3, f4, f6**	**f2, f3, f4, f6**	**f2, f3, f4, f6**	**f2, f3, f4, f6**
FCBF	Execution #1	f1, f2, f3, f4, f5, f6, f7, f8	f1, f2, f3, f4, f5, f6, f7, f8	f1, f2, f3, f4, f5, f6, f7, f8	f1, f2, f3, f4, f5, f6, f7, f8	f1, f2, f3, f4, f5, f6, f7, f8	**f1, f2, f3, f4, f5, f6, f7, f8**
	Execution #2	f1, f2, f3, f4, f5, f6, f7, f8	f1, f2, f3, f4, f5, f6, f7, f8	f1, f2, f3, f4, f5, f6, f7, f8	f1, f2, f3, f4, f5, f6, f7, f8	f1, f2, f3, f4, f5, f6, f7, f8	**f1, f2, f3, f4, f5, f6, f7, f8**
	Execution #3	f1, f2, f3, f4, f5, f6, f7, f8	f1, f2, f3, f4, f5, f6, f7, f8	f1, f2, f3, f4, f5, f6, f7, f8	f1, f2, f3, f4, f5, f6, f7, f8	f1, f2, f3, f4, f5, f6, f7, f8	**f1, f2, f3, f4, f5, f6, f7, f8**
	Execution #4	f1, f2, f3, f4, f5, f6, f7, f8	f1, f2, f3, f4, f5, f6, f7, f8	f1, f2, f3, f4, f5, f6, f7, f8	f1, f2, f3, f4, f5, f6, f7, f8	f1, f2, f3, f4, f5, f6, f7, f8	**f1, f2, f3, f4, f5, f6, f7, f8**
	Execution #5	f1, f2, f3, f4, f5, f6, f7, f8	f1, f2, f3, f4, f5, f6, f7, f8	f1, f2, f3, f4, f5, f6, f7, f8	f1, f2, f3, f4, f5, f6, f7, f8	f1, f2, f3, f4, f5, f6, f7, f8	**f1, f2, f3, f4, f5, f6, f7, f8**
	Intersection	**f1, f2, f3, f4, f5, f6, f7, f8**	**f1, f2, f3, f4, f5, f6, f7, f8**	**f1, f2, f3, f4, f5, f6, f7, f8**	**f1, f2, f3, f4, f5, f6, f7, f8**	**f1, f2, f3, f4, f5, f6, f7, f8**	**f1, f2, f3, f4, f5, f6, f7, f8**

Method	Execution	Set A	Set B	Set C	Set D	Set E	Intersection
SFS	Execution #1	f2, f3, f6, f8	f2, f3, f4, f6, f8	f1, f3, f6, f7, f8	f3, f6, f8	f1, f2, f3, f5, f6, f7	**f3, f6**
	Execution #2	f1, f3, f4, f6, f8	f1, f3, f4, f5, f6, f8	f1, f3, f4, f6, f8	f1, f2, f3, f6	f1, f3, f6	**f3, f6**
	Execution #3	f2, f3, f4, f5, f6, f8	f1, f3, f6, f7, f8	f1, f3, f6, f8	f2, f3, f6, f7, f8	f3, f4, f5, f6, f7, f8	**f3, f6, f8**
	Execution #4	f2, f3, f6, f8	f1, f3, f4, f5, f6, f8	f1, f2, f3, f4, f5, f6	f1, f3, f4, f5, f6	f1, f3, f6	**f3, f6**
	Execution #5	f3, f6, f7, f8	f1, f2, f3, f6	f2, f3, f4, f5, f6, f7, f8	f1, f3, f6, f8	f2, f3, f6, f8	**f3, f6**
	Intersection	**f3, f6, f8**	**f3, f6**	**f3, f6**	**f3, f6**	**f3, f6**	**f3, f6**
SFFS	Execution #1	f1, f3, f6	f2, f3, f4, f6	f1, f3, f4, f5, f6, f8	f1, f3, f6, f8	f3, f4, f6, f7, f8	**f3, f6**
	Execution #2	f1, f3, f5, f6, f7, f8	f3, f4, f6, f7, f8	f1, f2, f3, f4, f6, f8	f1, f3, f4, f5, f6, f8	f2, f3, f5, f6	**f3, f6**
	Execution #3	f2, f3, f4, f5, f6, f7, f8	f2, f3, f5, f6, f8	f1, f2, f3, f6, f7, f8	f2, f3, f6, f8	f2, f3, f6, f8	**f3, f6, f8**
	Execution #4	f3, f4, f6, f8	f1, f2, f3, f6	f3, f6, f7, f8	f3, f4, f6, f8	f3, f6, f8	**f3, f6**
	Execution #5	f2, f3, f4, f5, f6, f7	f1, f2, f3, f6, f7, f8	f2, f3, f4, f5, f6, f8	f1, f3, f6, f8	f3, f6, f7, f8	**f3, f6**
	Intersection	**f3, f6**	**f3, f6**	**f3, f6, f8**	**f3, f6, f8**	**f3, f6**	**f3, f6**
CI-SFFS	Execution #1	f1, f3, f4, f5, f6, f8	f1, f3, f4, f5, f6, f8	f2, f3, f6, f7, f8	f2, f3, f5, f6	f3, f5, f6, f8	**f3, f6**
	Execution #2	f1, f3, f4, f5, f6, f8	f1, f2, f3, f5, f6	f1, f2, f3, f5, f6	f1, f3, f5, f6	f1, f2, f3, f5, f6	**f3, f6**
	Execution #3	f1, f3, f4, f5, f6, f7	f1, f2, f3, f4, f6, f8	f1, f3, f4, f5, f6, f8	f1, f2, f3, f5, f6	f3, f5, f6, f8	**f3, f6**
	Execution #4	f1, f2, f3, f4, f6, f7, f8	f1, f3, f4, f5, f6, f8	f3, f4, f6, f7, f8	f1, f3, f4, f6, f7, f8	f2, f3, f5, f6, f7, f8	**f3, f6, f8**
	Execution #5	f1, f3, f5, f6, f7, f8	f1, f2, f3, f5, f6	f1, f3, f4, f6, f7, f8	f2, f3, f4, f6, f7, f8	f1, f3, f4, f6, f7, f8	**f3, f6**
	Intersection	**f1, f3, f6**	**f1, f3, f6**	**f3, f6**	**f3, f6**	**f3, f6**	**f3, f6**

It is shown that FCBF is shown to unable reduce the number of features, this is because this feature selection method is more suitable when handling high-dimensional data, because it analyze the correlation between features, which is feature relevancy and feature redundancy. Thus, this method will perform poorly when it failed to find the correlation between features, or they overestimate the correlation between features. In other domain of pattern recognition, the result obtained from FCBF can be considered as suboptimal result, however in this WI domain, this feature selection method is still considered to achieve the purpose of the experiment. This is because the purpose of feature selection in WI is not only to reduce the number of features; instead it is to determine the most significant features (unique features). Thus, FCBF considers all features are significant.

On the contrary, the rest of the methods (CFS, LVF, SFS, SFFS and CI-SFFS) are able to identify the significant features. It should be noted that the number of features selected is not always an indicator of a successful feature selection process. Therefore, further validation to justify the result produced by these methods must be designed, which is the classification accuracy.

It is also worth mentioning that although these feature selection methods yield different result with different size, they seem to always include the third feature (f3) in their results. Therefore, it can be concluded that the third feature (f3) is the most significant feature, and it is chosen as significant unique feature in order to proof the invarianceness of authorship in this work.

B. Classification Accuracy

The second measurement of this study is classification accuracy. The selected significant features from every feature selection methods must be justified and validated through identification performance. In order to justify the quality of feature subset produced by each method, the feature subsets are tested against classification, which uses MIC as the classifier. Table 3 is the result of classification accuracy for each feature subset.

These methods are both capable to identify the most significant features and at the same time they validate the invarianceness of authorship concept where the invariance between features for intra-class is lower than inter-class. The invarianceness of authorship is proven where the invarianceness between features using selected features for intra-class (same author) is smaller compared to inter-class (different author). This conforms the significant features is relate to invarianceness of authorship on WI.

Table 3. Experimental results on classification accuracy (%).

Method	Execution	Set A	Set B	Set C	Set D	Set E	Average
CFS	Execution #1	94.24	97.18	97.18	94.01	97.18	**95.95**
	Execution #2	94.24	97.18	97.18	94.01	97.18	**95.95**
	Execution #3	94.24	97.18	97.18	94.01	97.18	**95.95**
	Execution #4	94.24	97.18	97.18	94.01	97.18	**95.95**
	Execution #5	94.24	97.18	97.18	94.01	97.18	**95.95**
	Average	**94.24**	**97.18**	**97.18**	**94.01**	**97.18**	**95.95**
LVF	Execution #1	97.40	97.40	97.40	97.40	97.40	**97.40**
	Execution #2	97.40	97.40	97.40	97.40	97.40	**97.40**
	Execution #3	97.40	97.40	97.40	97.40	97.40	**97.40**
	Execution #4	97.40	97.40	97.40	97.40	97.40	**97.40**
	Execution #5	97.40	97.40	97.40	97.40	97.40	**97.40**
	Average	**97.40**	**97.40**	**97.40**	**97.40**	**97.40**	**97.40**
FCBF	Execution #1	98.08	98.00	98.06	97.57	97.62	**97.87**
	Execution #2	98.08	98.00	98.06	97.57	97.62	**97.87**
	Execution #3	98.08	98.00	98.06	97.57	97.62	**97.87**
	Execution #4	98.08	98.00	98.06	97.57	97.62	**97.87**
	Execution #5	98.08	98.00	98.06	97.57	97.62	**97.87**
	Average	**98.08**	**98.00**	**98.06**	**97.57**	**97.62**	**97.87**
SFS	Execution #1	97.40	97.18	96.92	96.14	96.94	**96.92**
	Execution #2	97.29	97.77	96.01	96.47	95.80	**96.67**
	Execution #3	97.63	97.30	95.78	96.80	97.05	**96.91**
	Execution #4	97.40	97.77	97.26	96.80	95.80	**97.01**
	Execution #5	97.51	96.59	97.38	96.14	96.49	**96.82**
	Average	**97.45**	**97.32**	**96.67**	**96.47**	**96.42**	**96.87**
SFFS	Execution #1	96.95	96.71	97.04	96.14	96.49	**96.66**
	Execution #2	97.40	97.18	96.58	97.13	96.94	**97.05**
	Execution #3	94.35	97.41	97.04	96.03	96.49	**96.26**
	Execution #4	97.06	96.59	96.58	96.14	96.03	**96.48**
	Execution #5	97.51	97.18	97.04	96.14	96.60	**96.89**
	Average	**96.66**	**97.02**	**96.85**	**96.32**	**96.51**	**96.67**
CI-SFFS	Execution #1	97.97	97.89	97.15	96.80	96.83	**97.33**
	Execution #2	97.85	97.06	97.26	96.91	97.17	**97.25**
	Execution #3	97.85	97.42	97.61	96.91	96.83	**97.32**
	Execution #4	97.97	97.89	96.92	97.13	97.39	**97.46**
	Execution #5	97.97	97.06	97.04	97.13	96.94	**97.23**
	Average	**97.92**	**97.46**	**97.19**	**96.98**	**97.03**	**97.32**

Based on the results, the accuracy is at its highest when the number of features is between 4-7 features. It is shown that FCBF produces the best accuracy (97.87%) and equal with the original dataset performance (97.87%). However, the number of features produced by FCBF is equal with the actual set (8 features). Meaning that, FCBF needs all features to produce the best performance.

The second best accuracy is LVF (97.40%). The results of LVF are shown to be stable, regardless of dataset and the number of execution. This is because the nature of the data that is consistent allows LVF to perform well. The next best accuracy is produced by CI-SFFS (97.32%). It is proven that although the time complexity has been greatly reduced, the classification accuracy has not been deteriorating; instead it is outperforming the classification accuracy of its predecessor (SFFS).

On the other hand, both SFS (96.87%) and SFFS (96.67%) with lower number of features still can obtain almost similar performance, although it is slightly lower than original dataset (97.74%). These feature selection methods outperform CFS. This is due to the behavior of these methods which can specifically identify the unique features in dataset, therefore it is resulting the highest performance. Besides that, the wrapper method is able to recognize importance of each feature in every iteration.

It is also shown that CFS is also capable to obtain good result (95.95%), although it is not as good as LVF, SFS and SFFS. Although FCBF is the enhancement of CFS, it is shown that CFS is still better than FCBF in some dataset. This is because FCBF determines the correlation between features faster than CFS, which may causing the method to overestimate the correlation between features, thus causing it to select all the features.

VI. Conclusions and Future Works

An extensive comparative study on feature selection methods for handwriting identification has been presented. This paper compared the merits of six different feature selection methods; three of them are filter methods, while the other three are wrapper method. An improved sequential forward floating selection has been developed to better adapt the nature of the data, and thus increase the performance. The exploration of significant unique features relates to authorship invarianceness has also been presented in this paper. A scientific validation has been provided as evidence of significant features can be used to proof the authorship invarianceness in WI.

The wrapper method is confirmed as the best option when it can be applied. In this paper, CI-SFFS is selected and used. When wrapper is not applicable, the results suggest using LVF. LVF produces the best results among other methods, both the number of features reduced and the classification accuracy. These results are produced by using commonly used feature selection methods, which is not purposely developed in handwriting identification domain.

Future works to hybridize the proposed feature selection method with recent optimization techniques is required. This is to allow better performance of the proposed method.

Acknowledgment

This work was supported in part by a Fundamental Research Grant Scheme FRGS/2010/FTMK/TK02/7-F00106 from the Ministry of Higher Education, Malaysia and Universiti Teknikal Malaysia Melaka, Malaysia.

References

[1] Dash, M., Liu, H. "Feature Selection for Classification", *Intelligent Data Analysis*, I, pp. 131-156, 1997.

[2] Saeys, Y., Inza, I., Larranaga, P. "A Review of Feature Selection Techniques in Bioinformatics", *Bioinformatics*, XXIII, pp. 2507-2517, 2007.

[3] Hall, M. A. "Correlation-based Feature Subset Selection for Machine Learning". *PhD Thesis*, University of Waikato, 1999.

[4] Gadat, S., Younes, L. "A Stochastic Algorithm for Feature Selection in Pattern Recognition", *Machine Learning Research*, VIII, pp. 509-547, 2007.

[5] Portinale, L., Saitta, L. "Feature Selection: State of the Art", in *Feature Selection*, Portinale, L., and Saitta, L. (eds.), Universita del Piemonte Orientale, Alessandria, 2002.

[6] Refaeilzadeh, P., Tang, L., Liu, H. "On Comparison of Feature Selection Algorithms". In *AAAI Workshop on Evaluation Methods for Machine Learning II*, pp. 34-39, 2007.

[7] Kudo, M., Sklansky, J. "Comparison of Algorithms that Select Features for Pattern Classifiers", *Pattern Recognition*, XXXIII, pp. 25-41, 2000.

[8] Schlapbach, A., Kilchherr, V., Bunke, H. "Improving Writer Identification by Means of Feature Selection and Extraction". In *8th International Conference on Document Analysis and Recognition*, pp. 131-135, 2005.

[9] Srihari, S.N., Cha S.-H., Lee, S. "Establishing Handwriting Individuality Using Pattern Recognition Techniques". In *6th International Conference on Document Analysis and Recognition*, pp. 1195-1204, 2001.

[10] Bin, Z., Srihari, S.N. "Analysis of Handwriting Individuality Using Word Features". In *7th International Conference on Document Analysis and Recognition*, pp. 1142-1146, 2003.

[11] Srihari, S.N., Huang, C., Srinivasan, H., Shah, V.A. "Biometric and Forensic Aspects of Digital Document Processing". In *Digital Document Processing*, Chaudhuri, B.B. (ed.), Springer, Heidelberg, 2006.

[12] Srihari, S.N., Cha, S.-H., Arora, H., Lee, S. "Individuality of Handwriting", *Forensic Sciences*, XLVII, pp. 856-872, 2002.

[13] Zhu, Y., Tan, T., Wang, Y. "Biometric Personal Identification Based on Handwriting". In *International Conference on Pattern Recognition*, pp. 797-800, 2000.

[14] Muda, A. K. "Authorship Invarianceness for Writer Identification Using Invariant Discretization and Modified Immune Classifier". *PhD Thesis*, Universiti Teknologi Malaysia, 2009.

[15] Schlapbach, A., Bunke, H. "Off-line Handwriting Identification Using HMM Based Recognizers". In *17th International Conference on Pattern Recognition*, pp. 654-658, 2004.

[16] Shen, C., Ruan, X.-G., Mao, T.-L. "Writer Identification Using Gabor Wavelet". In *4th World Congress on Intelligent Control and Automation*, pp. 2061-2064, 2002.

[17] Bensefia, A., Paquet, T., Heutte, L. "A Writer Identification and Verification System", *Pattern Recognition Letters*, XXVI, pp. 2080-2092, 2005.

[18] Yu, K., Wang Y., Tan, T. "Writer Identification Using Dynamic Features" in *Biometric Authentication*, Zhang, D., Jain, A.K., (eds.), Springer, Heidelberg, 2004.

[19] He, Z.Y., Tang, Y.Y. "Chinese Handwriting-based Writer Identification by Texture Analysis". In *2004 International Conference on Machine Learning and Cybernetics*, pp. 3488-3491, 2004.

[20] Muda, A. K., Shamsuddin, S. M., Darus, M. "Invariants Discretization for Individuality Representation in Handwritten Authorship". In *2nd International Workshop on Computational Forensics*, pp. 218-228, 2008.

[21] Pratama, S.F., Muda, A.K., Choo, Y.-H. "Feature Selection Methods for Writer Identification: A Comparative Study". In *2010 International Conference on Computer and Computational Intelligence*, pp. 234-239, 2010.

[22] Yu, L., Liu, H. "Efficient Feature Selection via Analysis of Relevance and Redundancy", *Machine Learning Research*, pp. 1205-1224, 2004.

[23] Geng, X., Liu, T.-Y., Qin, T., Li, H. "Feature Selection for Ranking". In *30th Annual International ACM SIGIR Conference*, pp. 407-414, 2007.

[24] Liu, H., Setiono, R. "A Probabilistic Approach to Feature Selection - A Filter Solution". In *International Conference of Machine Learning*, pp. 319-337, 1996.

[25] Yu, L., Liu, H. "Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution". In *20th International Conference on Machine Learning*, pp. 856-863, 2003.

[26] Whitney, A.W. "A Direct Method of Nonparametric Measurement Selection", *Computational*, pp. 1100-1103, 1971.

[27] Pudil, P., Novovicova, J., Kittler, J. "Floating Search Methods in Feature Selection", *Pattern Recognition Letters*, XV, pp. 1119-1125, 1994.

[28] Sun Microsystems, Inc. *Multithreaded Programming Guide*. Sun Microsystems, Inc, Santa Clara, 2008.

[29] Cajote, R.D., Guevara, R.C.L. "Global Word Shape Processing Using Polar-radii Graphs for Offline Handwriting Recognition". In *TENCON 2004 IEEE Region 10 Conference*, pp. 315-318, 2004.

[30] Parisse, C. "Global Word Shape Processing in Off-line Recognition of Handwriting", *Pattern Analysis and Machine Intelligence*, XVIII, pp. 460-464, 1996.

[31] Madvanath, S., Govindaraju, V. "The Role of Holistic Paradigms in Handwritten Word Recognition", *Pattern Analysis and Machine Intelligence*, XXIII, pp. 149-164, 2001.

[32] Steinherz, T., Rivlin, E., Intrator, N. "Offline Cursive Script Word Recognition – A Survey", *Document Analysis and Recognition*, II, pp. 90-110, 1999.

[33] Cheikh, F.A. "MUVIS: A System for Content-Based Image Retrieval". *PhD Thesis*, Tampere University of Technology, 2004.

[34] Vinciarelli, A. "A Survey on Off-line Cursive Word Recognition". *Pattern Recognition*, XXXV (7), pp. 1433-1446, 2002.

[35] Liao, S.X. "Image Analysis by Moment". *PhD Thesis*, University of Manitoba, 1993.

[36] Hu, M.K. "Visual Pattern Recognition by Moment Invariants", *Information Theory*, VIII, pp. 179-187, 1962.

[37] Alt, F.L. "Digital Pattern Recognition by Moments", *Journal of the ACM*, IX, pp. 240-258, 1962.

[38] Reiss, T.H. "The Revised Fundamental Theorem of Moment Invariants". *Pattern Analysis and Machine Intelligence*, XIII, pp. 830-834, 1991.

[39] Belkasim, S.O., Shridhar, M., Ahmadi, M. "Pattern Recognition with Moment Invariants: A Comparative Study and New Results". *Pattern Recognition*, XXIV, pp. 1117-1138, 1991.

[40] Pan, F., Keane, M. "A New Set of Moment Invariants for Handwritten Numeral Recognition". In *IEEE International Conference on Image Processing*, I, pp. 154-158, 1994.

[41] Sivaramakrishna, R., Shashidhar, N.S. "Hu's Moment Invariant: How Invariant Are They under Skew and Perspective Transformations?" In *Conference on Communications, Power and Computing*, pp. 292-295, 1997.

[42] Palaniappan, R., Raveendran, P., Omatu, S. "New Invariant Moments for Non-Uniformly Scaled Images", *Pattern Analysis and Applications*, III, pp. 78-87, 2000.

[43] Shamsuddin, S.M., Darus, M., Sulaiman, M.N. "Invarianceness of Higher Order Centralised Scaled-Invariants on Unconstrained Handwritten Digits", *Institute of Mathematics and Computer Sciences*, XII, pp. 1-9, 2001.

[44] Yinan, S., Weijun, L., Yuechao, W. "United Moment Invariant for Shape Discrimination". In IEEE *International Conference on Robotics, Intelligent Systems and Signal Processing*, pp. 88-93, 2003.

[45] Marti, U., Bunke, H. "The IAM-database: an English Sentence Database for Off-line Handwriting Recognition", *Document Analysis and Recognition*, V, pp. 39-46, 2002.

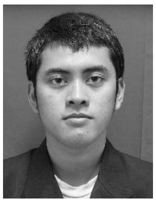
[46] Palhang, M., Arcot, S. "Feature Extraction: Issues, New Features, and Symbolic Representation". In *3rd International Conference on Visual Information and Information Systems*, pp. 418-427, 1999.

[47] Khotanzad, A., Lu, J.H. "Classification of Invariant Image Representations Using a Neural Network", *Acoustics, Speech and Signal Processing*, XXXVIII, pp. 1028-1038, 1990.

[48] Liu, C.-L., Dai, R.-W., Liu, Y.-J. "Extracting Individual Features from Moments for Chinese Writer Identification". In *3rd International Conference on Document Analysis and Recognition*, pp. 438-441, 1995.

[49] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I. H. "The WEKA Data Mining Software: An Update", *SIGKDD Explorations*, XI, pp. 10-18, 2009.

Author Biographies



Satrya Fajri Pratama was born in Bandung, Indonesia on October 11, 1988. He received his Bachelor of Computer Science in Software Engineering from the Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka. He is currently pursuing his Master of Science in Information and Communication Technology, also in Universiti Teknikal Malaysia Melaka. His research interests are including intelligent system, pattern recognition and software engineering.

Azah Kamilah Muda received the BSc and MSc from Universiti Teknologi Malaysia (UTM), Malaysia in 1997 and 1999 respectively. From 1997 to 2002 she joined the Department of Software Engineering at Faculty of Computer Science and Information System, Universiti Teknologi Malaysia as a lecturer. Since 2002 she has been with Universiti Teknikal Malaysia Melaka as a lecturer in Faculty of Information Technology and Communication. She obtained her PhD in bio-inspired pattern recognition from Universiti Teknologi Malaysia in 2009. Her research interests are Pattern Recognition, Image Processing, Soft & Biologically Inspired Computing, System Identification and Artificial Intelligence.

Yun-Huoy Choo was born in Muar, Johor, on 23rd February 1977. She obtained her Bachelor degree of Science and Computer with Education majoring in Mathematics (2000) and her Master of Information Technology (2002) from University of Technology Malaysia (UTM). She completed her PhD degree in 2008 from National University of Malaysia (UKM) specializing in Data Mining. She is currently lecturing in the Department of Industrial Computing, Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka (UTeM). Her research interest includes rough set theory, fuzzy sets theory, association rules mining, discretisation and feature selection, vagueness and uncertainty in specific, besides the fundamental and application of data mining in general.

Noor Azilah Muda is a lecturer and a researcher with 9 years of teaching and research experience in the Software Engineering area. Recently, Noor has a special interest in the Artificial Immune System (AIS) mechanism of recognizing cells that she later investigates the mechanism to recognize different patterns of music features for music genre classification. As her background is in the Software Engineering area, she is also involves in a research that investigates negotiations in requirement elicitation to improve the quality of requirements.