Proceedings of the 2009 IEEE Symposium on Computational Intelligence
in Security and Defense Applications (CISDA 2009)

Feature Selection and Classification in Genetic Programming: Application to Haptic-based Biometric data

Fawaz A. Alsulaiman, Nizar Sakr, Julio J. Valdés, Abdulmotaleb El Saddik, Nicolas D. Georganas

Abstract— **In this paper, a study is conducted in order to explore the use of genetic programming, in particular gene expression programming (GEP), in finding analytic functions that can behave as classifiers in high-dimensional haptic feature spaces. More importantly, the determined explicit functions are used in discovering minimal knowledge-preserving subsets of features from very high dimensional haptic datasets, thus acting as general dimensionality reducers. This approach is applied to the haptic-based biometrics problem; namely, in user identity verification. GEP models are initially generated using the original haptic biometric datatset, which is imbalanced in terms of the number of representative instances of each class. This procedure was repeated while considering an under-sampled (balanced) version of the datasets. The results demonstrated that for all datasets, whether imbalanced or under-sampled, a certain number (on average) of perfect classification models were determined. In addition, using GEP, great feature reduction was achieved as the generated analytic functions (classifiers) exploited only a small fraction of the available features.**

I. Introduction

THE integration of haptics into immersive virtual environments, has been an active research area the past decade. Immersive digital environments consist of computer-created scenes within which users can immerse themselves and interact with other users or various objects through a virtual reality experience. Conversely, haptic systems enable physical interactions with virtual three-dimensional objects through the sense of touch, and are therefore expected to become the next dimension of human-computer interaction. Haptic-based applications are wide, and span many areas, including medicine, rehabilitation, education and entertainment. In recent years however, the possible use of haptic devices in biometric systems has been suggested to enable improved user identification/verification performance over more traditional techniques, such as those based on hand-written signatures. Biometric systems provide a solution to ensure that protected services are solely accessible by a legitimate user. This is achieved while relying on users' behavioral and/or physiological characteristics. Conversely, haptic data depict trajectory, cutaneous as well as kinesthetic information which essentially consist of position, velocity, orientation, torque and force information, that are directly acquired from a haptic interface upon a user's interaction with a predefined virtual environment. However, the multidimensional and time-varying nature of the data renders haptic-based biometrics a challenging task, as the number of acquired features is enormous (in the thousands range). It is important to realize that this problem is not restricted to haptic-based biometrics, but in fact to any application that involves the analysis and interpretation of acquired haptic information to reveal certain patterns in the data. Consequently, this paper aims to investigate a technique to determine relevant attributes in high dimensional haptic-based datasets. The dataset considered in this study is generated using a haptic-enabled biometric application developed at DISCOVER laboratory, at the University of Ottawa.

Relevant feature selection and generation in high dimensional haptic-based biometric data is nearly unexplored in the literature. In [1], [2], Orozco *et al.* make use of the same dataset exploited in this paper in order to demonstrate the feasibility of a haptic-based user authentication system. The authors, however, distribute the high dimensional attributes of each signature across different instances, i.e. position, velocity, orientation, torque and force data acquired at time t_1 are assigned to instance $Inst_1$, data acquired at time t_2 are assigned to instance $Inst_2$, etc..., yielding a number of instances per user signature with only few attributes per instance. It is evident, however, that a more logical and adequate approach would be to assign all the generated haptic data attributes per signature to a single instance, i.e. each instance contains the entire (haptic-based) signature for a single user. This approach can lead to better analysis and interpretation of the haptic dataset, and improved discrimination between users. This, however, comes at the expense of having to deal with instances that possibly consist of thousands of attributes.

This paper explores the use of genetic programming, in particular gene expression programming (GEP), with the purpose of finding analytic functions that can act as superior classifiers in high-dimensional haptic feature spaces. More importantly, the generated explicit functions are used in discovering minimal knowledge-preserving subsets of features from the very high dimensional haptic datasets, thus acting as general dimensionality reducers. This approach is applied to the haptic-based biometrics problem; namely, in user identity verification.

The rest of the paper is organized as follows. In Section II the haptic data acquisition and preprocessing steps will be illustrated. In Section III haptic feature selection and

F. A. Alsulaiman, N. Sakr and A. El Saddik are with the School of Information Technology and Engineering, University of Ottawa, Canada (E-mail: fawaz@mcrlab.uottawa.ca; nsakr@site.uottawa.ca; abed@mcrlab.uottawa.ca).

J. J. Valdés is with the National Research Council Canada, Institute for Information Technology, Ottawa, Canada.

N.D. Georganas holds a Cátedra de Excelencia at the Univ. Carlos III de Madrid and is visiting researcher at IMDEA Networks, on leave from the School of Information Technology and Engineering, University of Ottawa.

classification using genetic programming will be discussed. In Section IV the experimental settings are provided. In Section V the experimental results are presented. Finally, conclusive remarks and some directions for future work are outlined in Section VI.

II. Haptic Data Acquisition and Preprocessing

In this section, the haptic-enabled virtual check application, as well as the acquired data will be described. Furthermore, techniques to solve the problem of imbalanced data sets will also be presented.

A. Haptic-enabled Virtual Environment

The experiments are performed using the Reachin Display [3], which integrates a haptic device with stereo graphics for an immersive and high quality 3D experience. The Reachin visuo-haptic interface enables users to see and touch virtual objects at the same location in space. This approach enables a superior integration of vision and touch than a conventional 2D screen-based display. The haptic stimulus is sensed using the SensAble PHANTOM Desktop force-feedback device, which is equipped with an encoder stylus that provides 6-degree-of-freedom single contact point interaction and positional sensing. In the case presented here, the visual stimuli consist of a virtual pen and a virtual check on which users can record their handwritten signature. The latter haptic-enabled virtual environment has been selected since handwritten signatures have been widely accepted as a mean to prove authenticity and authorship of a document. Conversely, the haptic stimuli are force and frictional feedback that attempt to mimic the tactile sensations felt when signing a traditional paper check. More specifically, the check is built on an elastic membrane surface with particular texture features, providing the users with a user-friendly and realistic feel of the virtual object. Moreover, similarly to conventional dynamic signature verification technologies, the virtual check application records a wide array of attributes that depict a user's physical and behavioral traits.

B. Haptic Datasets

The haptic-based handwritten signatures are diligently obtained from 13 different participants, where 10 signatures are collected per individual. A database is generated that is itself composed of a set of flat files (users' haptic-based signatures), that were collected on a workstation equipped with MS-OS 2000 and a XENON processor at the DISCOVER lab, at the University of Ottawa. The data acquired depict various distinct haptic features as a function of time. A number of haptic data types are considered that characterize the instantaneous state of the haptic system, including, three-dimensional position, force (pressure exerted on the virtual check), torque, and angular orientation. Furthermore, the multi-feature and multidimensional haptic data are recorded at 100 Hz. As the data is time-varying, the resulting number of attributes per signature is in fact the number of haptic data types considered (position, force, torque, ...) times the number of samples recorded per data type during each signature acquisition. This evidently leads to significantly large feature vectors that encompass thousands of haptic-based attributes.

C. Imbalanced Datasets

Imbalanced datasets occur in two class domains when the number of instances belonging to one class is significantly larger than the number of instances of the other class. In real life applications, it is not always possible to acquire the same number of instances for every class. This might occur due to a lack of sufficient knowledge about the minority class (the class containing only few instances), e.g. difficulties to collect information about rare species. Solving the problem of imbalanced datasets is of utmost importance as it can directly affect a classifier's performance. In fact, in such scenarios, classifiers can often predict the majority class with relatively high accuracy yet always misclassify the minority class; although in many data mining applications, such as in medical diagnosis domains, classification of the minority class is of crucial importance. Moreover, a classifier can reach a very high overall accuracy, however, still performs poorly when classifying the minority class. This can be observed from the following accuracy measure:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{1}$$

where TP, TN, FP and FN correspond to the true positive, true negative, false positive and false negative values respectively. In the case of a minority class where $TN + FP \gg TP + FN \geq 0$, i.e. TP and FN are relatively small values (which represent the minority class) in comparison to TN and FP which are associated with the majority class. Consequently, the corresponding accuracy measure will be misleading as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \approx \frac{TN}{FP + TN} \tag{2}$$

Many solutions have been proposed to tackle the problem of imbalanced data sets. A trivial solution is to resample the data by either over-sampling the minority class or under-sampling the majority class. The re-sampling techniques are either performed randomly or intelligently. In [4] Kubat and Matwin adopted Tomek's technique [5] where only instances of the majority class are removed to solve the imbalanced data sets problem. More specifically, it removes redundant and border line instances. They named their technique *One-Sided Selection* (OSS).

Chawla *et al.* [6] proposed *SMOTE*, an oversampling approach that over-samples the minority class with synthetic examples using k-nearest neighbors. The number of chosen nearest neighbors depends on the oversampling rate. In order to generate a synthetic sample, their technique initially chooses at random one of the k-nearest neighbors of a certain minority class sample. Then, it randomly generates a synthetic sample that falls on the line separating the two

genuine samples. In addition, the authors combined their approach with the randomly under-sampling (the majority class) technique. Pazzani *et al.* [7] suggested assigning costs to examples to reduce misclassification. The authors considered both the predicted classes and the actual classes in their costs assignment procedure. Japkowics and Stephen [8] performed several experiments that tackle the problem of imbalanced data sets. They concluded that the class imbalance problem is affected by four factors, namely the degree of class imbalance, the size of the training sets provided, the complexity of the resulting concept, and the type of classifier used. Hulse *et al.* [9] performed a comprehensive experiment on different solutions for the problem of imbalanced data sets using 35 real world bench mark data sets and 11 learning algorithms. In their experiments, the authors compared seven sampling techniques. It was observed that the performance of the sampling techniques depends directly on the learning algorithms used. Nevertheless, the authors ranked the random majority under-sampling as the sampling technique with best results followed by the random minority over-sampling method. The latter technique is performed by randomly replicating samples of the minority class based on the number of instances of the majority class. It was concluded that the two aforementioned simple sampling techniques performed much better than other intelligent schemes such as SMOTE [6] or OSS [4].

In this paper, we applied random under-sampling the majority class technique to solve the problem of imbalanced data sets. This method is initiated by omitting random instances from the majority class in order to obtain a balanced data set, i.e. a majority class with equal or relatively comparable number of instances to the minority class.

III. Haptic Feature Selection and Classification Using Genetic Programming

Analytic functions are among the most important building blocks for modeling, and consist of a classical form of expressing knowledge. In data mining, however, direct discovery of general analytic functions poses significant challenges due to the (in principle) infinite size of the search space. Within computational intelligence, Genetic Programming (GP) techniques are a promising approach to overcome this problem, as they aim at evolving computer programs, which ultimately are functions. There are many variants of GP algorithms in the literature; the one exploited in this work is the so-called Gene Expression Programming (GEP) [10], [11]. It is essentially an evolutionary algorithm as it uses populations of individuals, selects them according to fitness, and introduces genetic variation using one ore more operators.

GEP individuals are nonlinear entities of different sizes and shapes (expression trees) encoded as strings of fixed length. For the interplay of the GEP chromosomes and the expression trees (ET), GEP uses an unambiguous translation system to transfer the language of chromosomes into the language of expression trees and vise versa. The structural organization of GEP chromosomes allows a functional genotype/phenotype relationship, as any modification made in the genome always results in a syntactically correct ET or program. The set of genetic operators applied to GEP chromosomes always produces valid ETs.

Chromosomes in GEP itself are composed of genes structurally organized in a head and a tail [10]. The head contains symbols that represent both functions (elements from a function set F) and terminals (elements from a terminal set T), whereas the tail contains only terminals. Therefore, two different alphabets occur at different regions within a gene. For each problem, the length of the head h is chosen, whereas the length of the tail t is a function of h, and the number of arguments of the function with the largest arity.

As an example, consider a gene composed of the function set F=$\{Q, +, -, *, /\}$, where Q represents the square root function, and the terminal set T=$\{a, b\}$. Such a gene looks like (the tail is shown in **bold**): `*Q-b++a/-b`**`baabaaabaab`**, and encodes the ET which corresponds to the mathematical equation $f(a,b) = \sqrt{b} \cdot \left(\left(a + \frac{b}{a}\right) - ((a-b)+b)\right)$ simplified as $f(a,b) = \frac{b \cdot \sqrt{b}}{a}$.

Moreover, GEP chromosomes are usually composed of more than one gene of equal length. For each problem the number of genes as well as the length of the head has to be chosen. Each gene encodes a sub-ET and the sub-ETs interact with one another forming more complex multi-subunit ETs through a connection function. As an evolutionary algorithm GEP defines its own set of crossover, mutation and other operators [11]. Furthermore, to evaluate GEP chromosomes, different fitness functions can be used.

For the research described in this paper, GEP is exploited in an attempt to generate explicit analytic functions that can guarantee good discrimination capabilities between haptic-based handwritten signatures. Equally importantly, these functions are also used in discovering minimal knowledge-preserving subsets of attributes from the very high dimensional haptic-based biometric datasets, thus acting as general dimensionality reducers. The GEP-generated analytic functions are modeled as $y = f(v_1, \cdots, v_n)$, where $(v_1, \cdots, v_n)$ is the set of independent or predictor variables (attributes), and y the dependent or predicted variable (decision classes), so that $v_1, \cdots, v_n, y \in \mathbb{R}$, where $\mathbb{R}$ are the reals. In general terms, the model describing the program is given by $y = f(\vec{v})$, where $y \in \mathbb{R}$ and $\vec{v} \in \mathbb{R}^n$.

IV. Experimental Settings

The example high dimensional haptic dataset selected is that of [1], [2], and that were briefly described in Section II. They essentially consist of haptic-based handwritten signatures recorded from 13 different participants, where 10 signatures were collected per individual. In order to ensure accurate discrimination between the signatures, the obtained feature vectors were normalized to a common length of 10000. Essentially, the acquired haptic data types are re-sampled (upsampled/downsampled) when necessary to

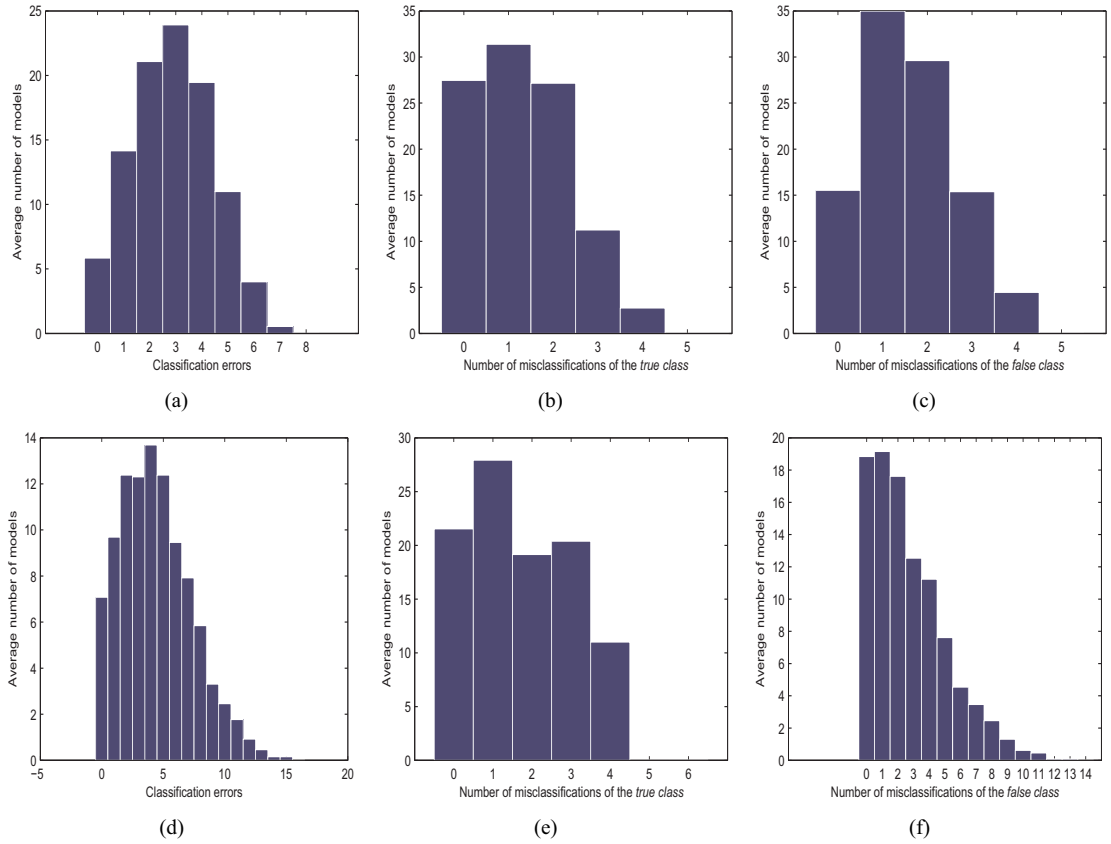

Fig. 1. Average number (across all 13 subsets) of generated models with respect to their classification performance when the following datasets are exploited: *undersampled_60%_datasets* and *imbalanced_60%_datasets*.

ensure a common feature vector length across all instances. The latter feature vector length was selected in such a manner to minimize the information loss that is most apparent when downsampling is performed. Consequently, the computed preprocessed dataset contains 130 instances, where each consists of 10000 features. However, in this paper we are concerned with biometric identity verification (as opposed to identification), which is a two-class classification problem (dichotomization), where a dichotomizer assigns class labels A (accept identity claim) or B (reject identity claim) to observed feature vectors x. The preprocessed dataset is therefore rearranged into 13 distinct datasets (one for each class), where in each only the instances of a single class are labeled *accept* ($A = 1$), whereas the remaining instances are labeled *reject* ($B = 0$). It is clear that the obtained datasets are highly imbalanced as the number of instances belonging to the *accept* class is much smaller than the number of instances associated with the *reject* class. Nonetheless, the obtained biometric identity verification datasets were then first divided into 60% training and 40% test sets, and also divided into 80% training and 20% test sets. Let's refer to the former datasets as *imbalanced_60%_datasets*, and the latter datasets as *imbalanced_80%_datasets*. The rearranged verification datasets were then under-sampled as illustrated in Section II-C in order to obtain a balanced version. The computed datasets, similarly to the imbalanced case, were then divided into 60% training and 40% test sets, and also divided into 80% training and 20% test sets. Let's refer to the former datasets as *undersampled_60%_datasets*, and the latter datasets as *undersampled_80%_datasets*.

The GEP experiments were performed with parameters fixed at the following suggested values [11]: population size = 1000, number of generations = 100000, genes/chromosome = 3, gene headsize = 8, constants = allowed (in $[0, 10]$), linking function = Addition, probabilities: inversion = 0.1, mutation = 0.044, istransposition = 0.1, ristransposition-prob = 0.1, onepointrecomb-prob = 0.3, twopointrecomb-prob = 0.3, generecomb-prob = 0.1, genetransposition-prob = 0.1, rnc-mutation= 0.01, dc-mutation-prob = 0.044, dc-inversion= 0.1, dc-istransposition = 0.1. In addition, the function set was very simple, composed only of basic arithmetic functions: $\{+, -, *, \div\}$.

V. Results

A series of multi- and single objective experiments were performed in order to investigate some of the properties of the data used within this study. For each of the 13 subsets in *imbalanced_60%_datasets*, 100 independent GEP runs were conducted, i.e. 100 different analytical functions are generated for the 1st subset, 100 different analytical functions are generated for the 2nd subset, ... It is important to mention that during each run, there are in fact 1000 different

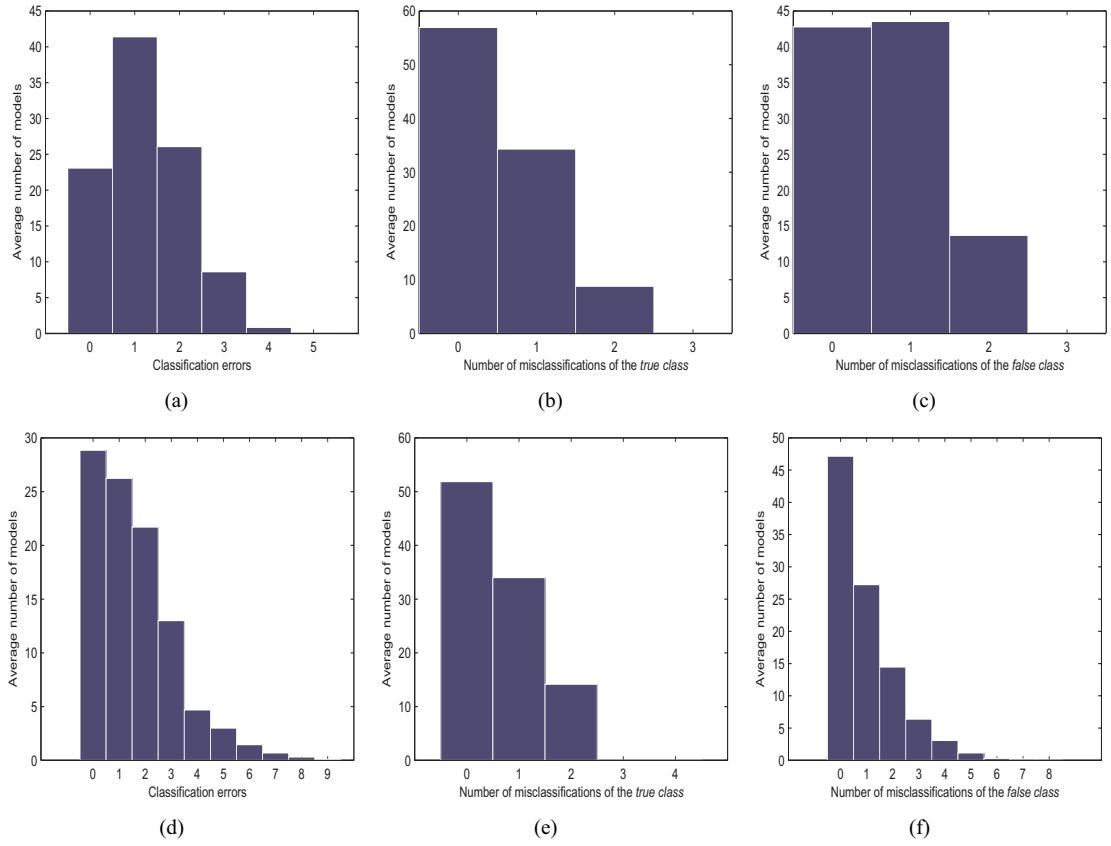

Fig. 2. Average number (across all 13 subsets) of generated models with respect to their classification performance when the following datasets are exploited: *undersampled_80%_datasets* and *imbalanced_80%_datasets*.

functions that are obtained (since the process is initiated with a population of size 1000), however, only the GEP model with the best discrimination performance is selected. This procedure is repeated with the three other computed datasets: *imbalanced_80%_datasets*, *undersampled_60%_datasets*, and *undersampled_80%_datasets*. An example of one of the analytic functions generated is illustrated in Equ. 3. More specifically, this model is associated with the 1st subset of *undersampled_60%_datasets*, i.e. it is essentially a classifier that considers that class 1 is the true (legitimate) identity, whereas all the other classes are false identities.

$$
\begin{aligned}
f(v_1, v_2, \cdots, v_{10000}) = \\
&((((v_{5047} + ((v_{2660} + v_{8848})/v_{6412}))/(v_{9277} + \\
&(v_{5147} * v_{8733}))) + ((((v_{9568} - v_{8608})/v_{3875}) + \\
&v_{5045}) - ((v_{8753} * v_{8685})/(v_{6054} + v_{2873})))) + \\
&((v_{3786} * ((v_{8460} + v_{1576})/v_{3911})) - \\
&((v_{1670}/v_{9280}) - v_{6762}))), \qquad (3)
\end{aligned}
$$

It can be observed that only few of the initial 10000 attributes are exploited in this equation. Consequently, as aforementioned, GEP can in this case be used to perform both feature selection, and generation of efficient classifiers. In Fig. 1 (a) and (d), two histograms are plotted that illustrate the average number (across all 13 subsets) of generated models with respect to their classification performance, i.e. the number of prediction errors, when *undersampled_60%_datasets* and *imbalanced_60%_datasets* are exploited, respectively. For example, in Fig. 1 (a), it can be shown that on average there are 6 models/subset that can achieve perfect verification results. Conversely, in Fig. 1 (b) and (e), two other histograms are plotted that illustrate the average number of generated models with respect to the number of misclassifications of the *true* class when *undersampled_60%_datasets* and *imbalanced_60%_datasets* are considered, respectively. These plots are necessary in order to present a fair comparison between the results obtained with the imbalanced and the undersampled datasets, as the former sets contain a significantly larger amount of instances that are associated with the *false* class (120 instances as opposed to 10 instances). Moreover, in Fig. 1 (c) and (f), two histograms are also shown that illustrate the average number of models generated with respect to their classification performance when *undersampled_60%_datasets* and *imbalanced_60%_datasets* are considered, respectively. However, in the latter case, only the number of misclassifications of the *false* class are considered. Similarly, in Fig. 2, the same results described above are generated for the following datasets: *undersampled_80%_datasets* and *imbalanced_80%_datasets*.

Many observations can be made from the results presented in Figs. 1 and 2. First, it can be seen that for all datasets, whether imbalanced or under-sampled, a certain number

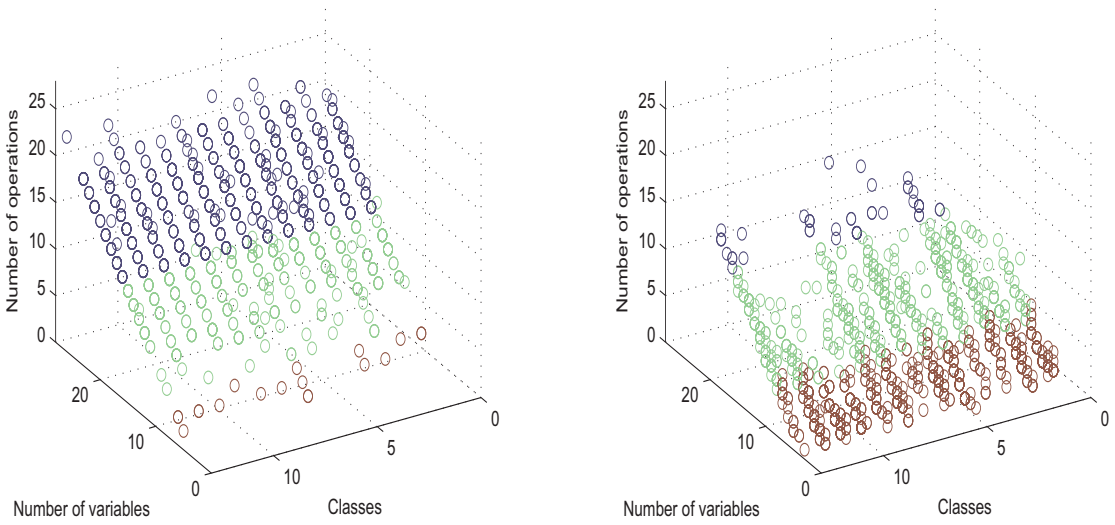

Fig. 3. Number of operations with respect to the number of variables present in the generated analytic functions, for each of the 13 classes in (a) *undersampled_60%_datasets* and (b) *imbalanced_60%_datasets*.

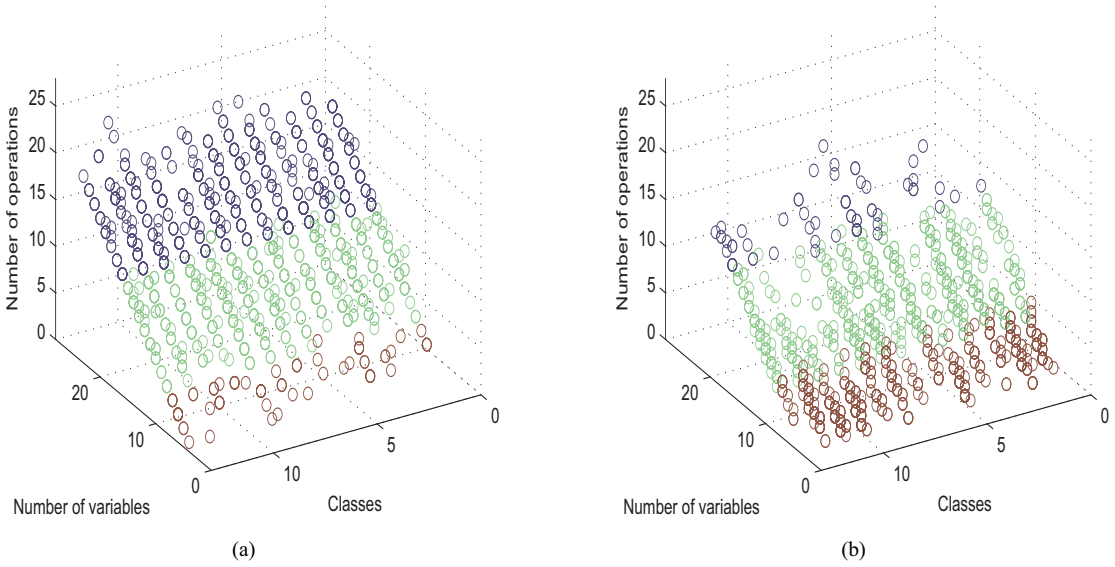

Fig. 4. Number of operations with respect to the number of variables present in the generated analytic functions, for each of the 13 classes in (a) *undersampled_80%_datasets* and (b) *imbalanced_80%_datasets*.

(on average) of perfect classification GEP models were achieved. Furthermore, from the histograms (a) and (d) of Figs. 1 and 2, it can be observed that a greater number of perfect verification models are obtained (with respect to the overall classification errors) when the imbalanced datasets are considered. However, since the presented performance results are simply an average of the number of generated models (across all 13 subsets) with respect to the classification errors, there are therefore no guarantees that perfect classifiers were computed for each class. A closer examination of the results revealed that perfect models were in fact generated for all 13 classes associated with the under-sampled datasets *undersampled_60%_datasets* and *undersampled_80%_datasets*; however, this was not the case for the imbalanced datasets. More specifically, perfect models were not achieved for classes 1, 4, 6, 9, and 13 when *imbalanced_60%_datasets* were exploited. Similarly, models with perfect classification performance were not achieved for class 13 when *imbalanced_80%_datasets* were used.

Furthermore, a larger number of perfect verification models were determined with the under-sampled datasets when the classification performance is regarded with respect to the number of misclassifications of the *true* class (see histograms (b) and (e) of Figs. 1 and 2). Therefore, as expected, classification performance of the minority class improves when the imbalanced datasets problem is resolved. Conversely, a

larger number of perfect GEP models are obtained with the imbalanced datasets when the classification performance is plotted with respect to the number of misclassifications of the *false* class (see histograms (b) and (e) of Figs. 1 and 2). This result can also be expected as GEP-based classifiers generated using the imbalanced datasets where trained using a much larger number of instances that belonged to the *false* class.

In Fig. 3 (a) and (b), a plot is presented to illustrate the number of operations with respect to the number of variables exploited in the generated GEP models for each of the 13 classes in *undersampled_60%_datasets* and *imbalanced_60%_datasets*, respectively. It is, however, important to mention that the *number of variables* refers to the number of distinct variables used in the explicit equations (duplication of identical variables are not counted). Conversely, the *number of operations* includes all arithmetic operations used in each function (duplication are counted). The parameters were selected as such in order to demonstrate simultaneously, the complexity of the computed GEP models as well as their feature reduction capabilities. Furthermore, the models included in the plots are only those with which 100% training accuracy was achieved. Similarly, Fig. 4 (a) and (b) illustrate the number of operations with respect to the number of variables present in the generated analytic functions, for each of the 13 classes in *undersampled_80%_datasets* and *imbalanced_80%_datasets*, respectively. It is evident that models with the fewest number of operations and variables are more desirable. It can be seen that in Figs. 3 and 4, for both imbalanced or under-sampled datasets, a relatively large number of the generated models (analytical functions) are in fact low in complexity as only a few number of operations are performed. Moreover, almost all the generated classifiers use only a fraction of the 10000 attributes initially introduced to the GEP algorithm.

VI. Conclusions

The genetic programming approach exploited in this study, in particular gene expression programming, proved to be very effective for generating analytic models that can simultaneously serve two important functions: behave as classifiers in high-dimensional haptic feature spaces, and act as general dimensionality reducers. The obtained experimental results are very promising, but preliminary. A more thorough experimental study of this approach is necessary in order to assess the technique's general behavior when applied in haptic-based biometrics. For example, it would of interest to analyze whether any overall improvements would be achieved if the haptic datasets are over-sampled, as opposed to under-sampled as it was performed in this work, when attempting to overcome the class imbalance problem.

References

[1] A. El Saddik, M. Orozco, Y. Asfaw, S. Shirmohammadi and A. Adler, "A Novel Biometric System for Identification and Verification of Haptic Users," IEEE Transactions on Instrumentation and Measurement, vol. 56, no. 3, pp. 895-906, 2007.

[2] M. Orozco, M. Graydon, S. Shirmohammadi, and A. El Saddik,"Experiments in Haptic-Based Authentication of Humans," Journal of Multimedia Tools and Applications, vol. 37, no. 1, pp. 73-92, 2007.

[3] *Reachin Technologies ab. Reachin Display.* http://www.reachin.se/products/.

[4] M. Kubat and S. Matwin, "Addressing the Curse of Imbalanced Training Set: One Sided Selection," in the Proceedings of the Fourteenth International Conference on Machine Learning, pp. 179-186, Nov. 2004.

[5] L. Tomek, "Two Modifications of CNN," IEEE Transactions on Systems, Man and Communications, vol. 6, no. 11, pp. 769-772, 1976.

[6] N. V. Chawla, K. W. Bowyer, W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," Journal of Artificial Intelligence Research, vol. 16, pp. 321-357, 2002.

[7] M. Pazzani, C. Merz, P. Murphy, K. Ali, T. Hume, and C. Brunk, "Reducing Misclassification Costs," in the Proceedings of the 11th International Conference of Machine Learning, pp. 217-225, 1994.

[8] N. Japkowics, and S. Stephen, "The Class Imbalance Problem: A Systematic Study," Intelligent Data Analysis Journal, vol. 6, no. 5, pp. 429-449, Nov. 2002.

[9] J. V. Hulse, T. M. Khoshgoftaar, and A. Napolitano, "Experimental Perspectives on Learning from Imbalanced Data," in the Proceedings of the 24th International Conference on Machine Learning, pp. 935-942, 2007.

[10] C. Ferreira, "Gene Expression Programming: A New Adaptive Algorithm for Problem Solving," Journal of Complex Systems, vol. 13, no. 2, pp. 87-129, 2001.

[11] C. Ferreira, *Gene Expression Programming: Mathematical Modeling by an Artificial Intelligence*. Springer Verlag, 2006.