Noname manuscript No.
(will be inserted by the editor)

Feature Selection for Speaker Verification using Genetic Programming

Róisín Loughran · Alexandros Agapitos · Ahmed Kattan · Anthony Brabazon · Michael O'Neill

Received: date / Accepted: date

Abstract We present a study examining feature selection from high performing models evolved using Genetic Programming (GP) on the problem of Automatic Speaker Verification (ASV). ASV is a highly unbalanced binary classification problem in which a given speaker must be verified against everyone else. We evolve classification models for 10 individual speakers using a variety of fitness functions and data sampling techniques and examine the generalisation of each model on a 1:9 unbalanced set. A significant difference between train and test performance is found which may indicate overfitting in the models. Using only the best generalising models, we examine two methods for selecting the most important features. We compare the performance of a number of tuned machine learning classifiers using the full 275 features and a reduced set of 20 features from both feature selection methods. Results show that using only the top 20 features found in high performing GP programs led to test classifications that are as good as, or better than, those obtained using all data in the majority of experiments undertaken. The classification accuracy between speakers varies considerably across all experiments showing that some speakers are easier to classify than others. This indicates that in

This work was carried out as a collaboration of projects funded by Science Foundation Ireland under grant Grant Numbers 08/SRC/FM1389 and 13/IA/1850.

R. Loughran, A. Agapitos, A. Brabazon, M. O'Neill
Natural Computing Research and Applications Group (NCRA)
University College Dublin, Ireland
Tel.: +353-1-7198038
E-mail: roisin.loughran@ucd.ie, alexagapitos@gmail.com, anthony.brabazon@ucd.ie, m.oneill@ucd.ie

A. Kattan
Computer Science Department
Um Al-Qura University
Saudi Arabia
E-mail: kattan.ahmed@gmail.com

such real-world classification problems, the content and quality of the original data has a very high influence on the quality of results obtainable.

Keywords Speaker Verification · Feature Selection · Unbalanced Data · Genetic Programming

1 Introduction

Speaker recognition is the process of identifying a person from their voice. Each individual's voice is audibly unique due to physical attributes such as length of vocal tract, size of larynx etc. along with habitual characteristics such as accent and inflection. Whereas Automatic Speaker Recognition (ASR) involves recognising one speaker from many, Automatic Speaker Verification (ASV) is the process of accurately verifying that a speaker is who they claim to be. Thus for any given speaker, ASV is a binary classification problem, either the subject is in the speaker class or the non-speaker class. This may be posed as a text-dependent or text-independent problem. Text-independent ASV has important applications in the fields of phone banking, shopping and security systems. Such systems should be able to determine with 100% accuracy from any text-independent utterance if a person speaking is who they claim to be. The proposed study considers text-independent ASV as a highly unbalanced classification problem.

In real-world classification problems, it is common for datasets to be biased towards one class, resulting in an unbalanced class distribution. In unbalanced binary classification, one class has a small number of instances (the *minority* class) whereas the other contains a much larger number of data instances (the *majority* class). Such distributions are common in realistic data. ASV is inherently a highly unbalanced binary-classification problem since it requires accurately recognising one speaker from everyone else. In ASV, the minority class contains examples from the to-be-verified speaker, whereas the majority class contains examples from the rest of the speakers (i.e. impostors). This imbalance in class distribution is a significant problem as it introduces a learning bias and often results in classification models that are not accurate in the cases of the to-be-verified speaker. In general, in unbalanced classification problems, the smaller the ratio of minority class examples to majority class examples, then the stronger this bias becomes and the harder it is for a classifier to generalise [7]. Although classic machine learning studies rarely took this imbalance into account, there has been an increasing interest in machine learning with unbalanced data since 2000 [3,4,11].

1.1 Contributions of this research

The results in this paper are an extension of work presented at EvoIASP 2016 [50]. This work was carried out as an investigation on the application of GP to the problem of ASV, specifically with a very high class imbalance. With the notable exception of [15], ASV is an application area that has received little attention from the GP community. The 1:9 class imbalance proposed in this work is distinctly higher than any other studies, both in ASV studies on GP such as [15], but also in the more general literature on unbalanced classification with GP such as those described in Section 2. The simulations were performed on the TIMIT corpora [24], a regularly-used dataset for speaker recognition and verification. We used the original, noiseless TIMIT corpora to study the generalisation of GP-evolved programs on ASV. A number of different methods for cost-sensitive training and data sampling were compared in terms of their effectiveness to assist with the evolution of good-generalising programs. The details of these systems, the data used and the full original set of features is given in Section 3. From this initial study, we found that GP performed well but a decrease in classification accuracy between training and test results indicated a tendency for the program trees to overfit the data.

This paper progresses with this work by comparing the GP performance with four alternative machine learning classifiers, namely a Support Vector Machine, Random Forest, Logistic Regression and Gradient Boosting Classifier. The results of these experiments are given in Section 4.2. In general, features chosen for such classification experiments are those found in the literature; few studies offer experimental reasons as to their choice of features to include. This is because feature selection is a daunting task; the user must reduce the dimensionality of the search space with minimal loss of information. In recent years, evolutionary computational approaches have been applied to feature selection with promising results [65]. They note however that compared with evolutionary approaches such as genetic algorithms or particle swarm analysis, there are a much smaller number of works using GP for feature selection. The aim of this proposed paper is to use the evolved GP trees from our previous experiments to inform these machine learning algorithms of suitable features to use for ASV. For this purpose, we analysed the terminal nodes of highly-performing trees using two separate feature selection methods, termed the Original Selection and Accuracy Selection methods, to determine the most useful features from the total 275 used in the terminal set. These feature selection methods are used in an attempt to optimise the performance of the four independent machine learning classifiers. A comparison of classification results with the inclusion of all 275 features against the performance using only the top 20 features from both feature selection methods is provided. A description of these results, including statistical analysis, is given in Section 4.3. Finally we perform a comparison between the speakers used throughout the experiments and discuss the differences between these experiments and previous speaker verification experiments in Sections 4.4 and 4.5.

2 Background

This section provides an overview of traditional classification models for ASV and describes the two main categories of methods for tackling class imbalance problems. We specifically look at experiments from the GP literature that have tackled the class-imbalance problem.

2.1 Automatic speaker verification

Standard ASV depends on extracting features from a given set of speech samples and using this set of features to train and test a given classifier. The first method of implementing speaker models was based on Vector Quantisation (VQ) [41]. VQ determines a standard measure such as the Euclidean distance, known as the average quantisation distortion, between the feature vector from a speaker $X = \{x_1, \ldots, x_T\}$ to that of a reference vector $R = \{r_1, \ldots, r_K\}$, where T is close or ideally equal to K. This idea of representing speech with vectors has been expanded with the development of *Supervectors*. These Supervectors in general consist of any high and fixed dimensional representation of an utterance. One of the most prominent classification methods for ASV to emerge in the early 1990s were those based on Gaussian Mixture Models [57]. Such classification models are based on the distribution of features from a section of speech by a Gaussian mixture density. Over the next three decades other classification techniques were applied to the problem of speaker recognition and verification such as Supervectors, Support Vector Machines (SVM) [9], Artificial Neural Networks [59], ensemble learning [48], and Genetic Programming [15].

A number of studies focussed on methods to counteract inter-speaker and inter-session variability by examining channel compensation between recordings. Such studies have used feature mapping to transform obtained features into a channel-independent feature-space. Methods such as Joint Factor Analysis [37], i-vectors [38] and PDLA [18] were applied for this purpose. Recent studies have reported good results by extending the i-vector representation to consider utterances of arbitrary duration [39], the total variability space [14] and local session variability [12]. This focus on channel effects is in part driven by the NIST Speaker Recognition Evaluation [1] which evaluates novel speaker recognition systems on a corpora of phone recordings. The contest has become so competitive that large scale collaborations have been founded to propose multiple submissions to the challenge with promising results and conclusions [58]. The proposed study uses noiseless original speech data, rather than the noisy channel dependent recordings used for these experiments.

Spoofing or imposture is a well-known problem in biometric systems [23]. In recent years, studies have been undertaken to consider the problem of spoofing in ASV [43,2,23,64,63]. There are four identified typical spoofing techniques for ASV: Impersonation, Speech Synthesis, Replay and Voice Conversion. Even with this new interest, it is noted in [64] that spoofing in the domain of ASV is still in its infancy. They point to the use of non-standard databases, protocols and metrics giving rise to difficulties in comparable research and countermea-

[1] http://www.nist.gov/itl/iad/mig/ivec.cfm

sures that lack generalisation. The focus of the proposed work is towards generalisation by using unbalanced data and comparison of selected features rather than a specific focus on countermeasures of a specific spoofing technique.

Most recent studies have used a dimensional reduction of previous features such as Acoustic Factor Analysis [28], Multitaper windows [43,55], or a combination of features and classification techniques in an attempt to develop a robust speaker verification system [61,26,46]. The breadth of methods still being used demonstrate the difficulty of the problem. The current study poses noiseless ASV as an unbalanced binary classification problem and considers how feature selection may be used to improve the performance of a number of classifiers on this problem.

2.2 Unbalanced data

In problems such as ASV we are trying to verify one speaker from everyone else, inherently an unbalanced problem. There exist a number of methods for learning good-generalising classifiers for class imbalance datasets. The taxonomy of these methods originally consisted of two major categories; those of *training data sampling* (often called external methods) and *cost-sensitive training* (internal methods). The work of [7] provides an excellent overview of these methods, with references from both statistical machine learning and GP. A brief overview of both approaches is provided below.

2.2.1 Training-data sampling.

Balancing of training examples can be achieved either by *over-sampling* the minority class or *under-sampling* the majority class [5]. These sampling methods have limitations. Under-sampling can reduce training times as it reduces the number of training samples, but it is possible that this reduction leads to a loss of important information and therefore loss of generalisation. Over-sampling increases training times and can also lead to overfitting due to the repetition of samples [47]. The relative benefits of under-sampling versus over-sampling appear to be problem-dependent [21,35]. Synthetic over-sampling and editing have been often shown to be superior to the sampling techniques described above. Synthetic over-sampling of the minority class creates additional examples by interpolating between several similar examples [6], while *editing* removes noisy or atypical examples from the majority class [44].

The work of [35] found that random over-sampling is more effective than random under-sampling for C5.0 decision-tree learning, whereas it was shown that under-sampling outperforms over sampling for the C4.5 decision tree learner with an unbalanced dataset in [21]. As part of their study into the nature of the class imbalance problem, [35] found that the higher the degree of class imbalance, the higher the complexity of the concept of imbalance, and the smaller the size of the overall training set the greater the effect of class imbalance in classifiers' sensitivity to the problem.

2.2.2 Cost-sensitive training.

In a typical classification problem, we are given a training set of N examples $\{(x_i, y_i)\}_{i=1}^{N}$, where $x \in \mathbb{R}^d$ is a d-dimensional vector of explanatory variables and $y \in C = \{1, \ldots, c\}$ is a categorical response variable, with joint distribution $P(x, y)$. We seek a function $f(x)$ for predicting y given the values of x. The loss function $L(y, f(x))$ for penalising errors in prediction can be represented by a $K \times K$ cost matrix L, where $K = card(C)$. L will be zero on the diagonal and non-negative elsewhere, where $L(k, l)$ is the price paid for misclassifying an observation belonging to class C_k as C_l. Most often, in cases of balanced datasets, a *zero-one* loss function $L(y, f(x)) = I(y \neq f(x))$ [2] is used, where all misclassifications are charged one unit. In the case of unbalanced datasets, the cost matrix can be adjusted to increase the cost of misclassifying the examples of the minority class.

2.3 GP on unbalanced datasets

Genetic Programming has been applied to unbalanced datasets in a number of studies. Work using the data sampling techniques of Random Sampling Selection (RSS) and Dynamic Subset Selection (DSS) is reported in [13,25]. In [13] a two-level sampling approach is first used to sample blocks of training examples using RSS and then select examples from within those blocks using DSS. In [25] DSS is used to bias the selection of training examples towards hard-to-classify examples, while RSS was used to bias towards the selection of minority class training examples.

Cost adjustment strategies usually focus on adapting the fitness function to reward programs which have good accuracy on both classes with better fitness, while penalising those with poor accuracy on one class with low fitness. The use of different misclassification costs to incorrect class predictions is reported in [33]. In the work of [22] an adaptive fitness function increases misclassification costs for difficult-to-classify examples. In [60] RSS and DSS are used in conjunction with three novel fitness functions with an application to a network intrusion detection problem. The work of [54] used both rebalancing of data and cost-sensitive fitness functions in comparing GP with other data-mining approaches to predict the rate of student failure in school. The work of [7] used six datasets with different class imbalance ratios and applied GP with a number of different fitness functions. A multi-objective GP approach for evolving accurate and diverse ensembles of GP classifiers that perform well on both minority and majority classes was proposed in [8]. A weighted average composed of error rate, mean squared error and a novel measure of class separability similar to Area Under Curve is used in [62]. In the work of [20], data sub-sampling is used in combination with the average of the geometric mean between minority and majority class accuracies and the Wilcoxon-Mann-Whitney statistic.

[2] $I(\cdot)$ is the indicator function.

2.3.1 GP on unbalanced ASV

To the best of our knowledge, the application of GP to speaker verification has only been reported in the work of [15]. One of the principal applications of ASV systems is remotely confirming the identity of a person for reasons of security such as telephone banking. The literature review conducted in [15] showed that while good results have been reported using a variety of statistical machine learning systems on noiseless input signals, most systems suffer heavily if the signal is transmitted over a noisy transmission path (i.e. a telephone network). In order to create a "noisy" environment, several datasets were derived from the original TIMIT corpora using filters that included both additive and convolutive noise. GP experiments were set to evolve classifiers based on extracted features impaired by noise. Twenty-five speakers to-be-verified and forty-five "impostors" were selected from the TIMIT corpora. For each of the to-be-verified speakers the training set consists of fifteen seconds of to-be-verified speech and forty-five seconds of impostor speech. This results in a minority class to majority class imbalance ratio of 1:3. A pool of hand-engineered features were extracted from the raw signal to populate the terminal set. The fitness function was dynamically biased to concentrate on difficult-to-classify examples. Finally, an island model was used to improve population diversity. Results showed that generated programs can be evolved to be resilient to noisy transition paths, which was mainly attributed to the speaker-dependent and environment-specific feature selection inherent in GP.

Although both the proposed study and the paper described above discuss GP applied to the problem of ASV, they differ from one another significantly in their use of the data and in their intended goals. One focus of the proposed work lies in viewing ASV as a highly unbalanced classification problem. Studies such as [7] consider unbalanced data problems directly in relation to the level of imbalance present in the data, while ASV problems (including the one discussed above) rarely consider a ratio above 1:3. The proposed study pushes this imbalance to a much more severe 1:9 ratio of minority to majority class, thus making it a substantially more difficult classification problem. While [15] used filtered noise to deteriorate the signal as may be expected in a real-world scenario, the focus of the proposed study is to highlight two alternative real world problems in ASV: that of reducing the amount of data used to represent the signal (through feature selection) and of considering the high number of imposter classes that a real world system would be subject to (through a very high class imbalance). Typical features from the literature were implemented in [15] which led to an interesting discussion on which features were chosen by their successfully evolved programs — but specifically in relation to which were useful for a noisy vs. a noiseless problem. Furthermore they refrained from actually implementing any feature selection methods. An alternative interesting paper by the same authors [16] actually focussed on the idea of creating a *Superfeature* for audio classification using GP. This study did not directly tackle the problem of ASV however, but instead trained a system to classify between three different types of audio namely noise, speech and music. The proposed study actually implements two Feature Selection techniques developed from the original GP runs and uses these in an attempt to increase the performance of a bank of separate ML classifiers. As such, the focus of the study is taken away from the GP classification performance and towards its performance as a meaningful method of feature selection or data reduction.

3 Methods

3.1 Speaker Corpus

The speech recordings used in this study are taken from the TIMIT corpora [24]. This was chosen due its regular use in the speaker recognition and verification literature. The corpora consists of 630 speakers, 192 female and 438 male, from 8 American dialects each reading 10 phonetically rich sentences. Each sentence was recorded on a high quality microphone at a sampling rate of 16kHz.

3.2 Training and Test data

For these experiments we chose 10 random speakers, 4 female and 6 male, from the corpus and developed a classifier for each speaker. For each experiment the audio from the given speaker is the to-be-verified minority class and the audio from the nine other speakers constitute the majority class. In this manner we created a 1:9 class imbalance ratio for each experiment.

Each speaker offers 10 utterances of approximately 3 seconds each. To increase the number of speech utterances, we split each sentence into three equal parts of approximately 1 second. Early analysis showed that the third part of each sentence was of lower timbral quality than the preceding sections, possibly due to pausing or hesitation at the end of the utterance. Thus only the first two thirds of each sentence were included in the learning dataset of 200 examples. In the experiments, a *training set* of size 120 examples is used to evolve programs, and a *test set* of 80 examples is used to assess generalisation. When splitting the data into training and tests sets we ensure using stratification that the class imbalance ratio of 1:9 is maintained in both sets.

The features calculated on this data are detailed in Section 3.4. As audio is a time-varying signal, many of these features are measured across the duration of the signal giving multiple points for each feature. Rather than reducing these features using the statistical mean or variance of the windowed signal, we employed Principal Component Analysis (PCA) on these time-varying, high-dimensional features. PCA was used on these results to record the maximum variance within each feature while reducing the dimensionality of the data. Each time we employed PCA, we recorded the top four principal components for that particular feature. In total this resulted in 275 features calculated on 200 data samples.

3.3 GP systems

A number of systems tailored to unbalanced classification problems from the literature were chosen for this study. These are detailed below.

3.3.1 ST.

This *Standard* system is trained using the original unbalanced dataset. We employ a version of the MSE-based loss function that has been shown [7] to improve upon the performance of fitness functions based on classification accuracy [3] or the weighted average of true positive and true negative rates. Given N training examples $\{(x_i, t_i)\}_{i=1}^{N}$ containing the examples of both majority and minority classes, L_{MSE} is defined as:

$$L_{MSE} = \frac{1}{N}\sum_{i=1}^{N}(\Phi(f(x_i)) - t_i)^2 \tag{1}$$

where

$$\Phi(x) = \frac{2}{1+e^{-x}} - 1 \tag{2}$$

and $f(x_i)$, t_i are the program output and target values for the i^{th} training case respectively. The sigmoid function in Equation 2 scales $f(x)$ within the range $\{-1\ldots1\}$. Similarly to [7], the target value for the majority class is set to -0.5, while the target value for the minority class is set 0.5. Classification is based on a zero-threshold approach; positive program output is mapped to the minority class label, while negative output is mapped to the majority class label.

3.3.2 AVE.

This *Average* system is trained using the original unbalanced dataset. The loss function uses a weighted-average classification accuracy of the minority and majority classes [7]. Minority accuracy corresponds to the true positive rate, whereas majority accuracy corresponds to true negative rate. The weighting coefficient between the two is $0 < w < 1$. When w is set to 0.5, the accuracy of both classes contributes equally to the loss function. In case of $w > 0.5$ the accuracy of the minority class contributes more to the loss function, lowering the contribution of the majority class accuracy. The loss function L_{AVE} is defined as:

$$L_{AVE} = 1.0 - \left(w \times \frac{TP}{TP+FN} + (1-w) \times \frac{TN}{TN+FP}\right) \tag{3}$$

where TP, TN, FN, FP is the count of true positives, true negatives, false negatives and false positives respectively.

[3] The number of examples correctly classified as a fraction of the total number of training examples.

3.3.3 US.

The *Under-sampling* system adjusts the number of examples in the majority class to balance it with the size of the minority class. Since static under-sampling of the majority class examples can introduce unwanted sampling bias and discard potentially useful training examples, we resort to a dynamic version of under-sampling. At every generation, a new set of examples is drawn random-uniformly from the set of training examples of the majority class. Under-sampling ensures that the number of examples drawn from the majority class is the same as the number of examples for the minority class. The loss function used is given in Equation 1.

3.3.4 RS.

This *Random-sampling* method is implemented as an extreme form of under-sampling. A type of random sampling technique in which programs are evaluated on a *single* example drawn uniform-randomly from the entire training dataset in each generation was shown to improve the generalisation of programs as compared to the use of the complete training set [27]. An obvious extension of this method to datasets with class imbalance is to populate the training set with *two* randomly-drawn examples (different in each generation), one each from the minority and majority class. The loss function used is given in Equation 1.

3.4 Features

In the context of speaker verification, a ‘feature’ is any numerical measurement that can be used to describe or distinguish a given speaker. High-level prosodic features such as those describing fundamental frequency or rhythm have been used for speaker verification [17]. Such high-level features are difficult to measure accurately, possible to mimic and are susceptible to the speaker’s emotions. Thus lower level spectral, cepstral, spectro-temporal and statistical features are more common in verification tasks. The short-term spectral features used in this study are described below. These features were chosen due to their frequent use in the literature [41].

Mel-frequency Cepstral Coefficients MFCCs have become the standard measure of speech analysis for some time [56]. They consist of a set of coefficients that can represent the spectral quality within a sound according to a scale based on human hearing. Obtaining the MFCCs consists of windowing the sound, calculating amplitude spectrum of cepstral feature vector for each frame and then converting this to the perceptually derived *mel-scale* [49]. As described earlier, the dimensionality of the MFCCs were reduced in this study using PCA. The first four PCs of the first 12 MFCCs along with their derivatives are included in these experiments resulting in 48 discrete MFCC feature values.

Linear Prediction Coefficients Linear prediction calculates a given signal based on a linear combination of the previous inputs and outputs [53]. As a spectrum estimation it offers good interpretation in both the time and frequency domains. In the time domain, LP predicts according to

$$s[\tilde{n}] = \sum_{k=1}^{p} a_k s[n-k] \tag{4}$$

where $s[\tilde{n}]$ is the predicted signal, $s[n]$ is the observed signal and a_k are the predictor coefficients at time step n and p is the order of the system. The prediction error or residual is defined as the difference between the predicted signal and the observed signal:

$$e[n] = s[n] - s[\tilde{n}] \tag{5}$$

The linear predictive coefficients (LPCs), a_k, are determined by minimising this residual. This analysis leads to the *Yule-walker equations* that can be efficiently solved using *Levinson-Durbin recursion* [34]. Given the LPC coefficients a_k, k = 1 . . . p, the linear predictive cepstral coefficients (LPCCs) are computed using the recursions [40]:

$$c[n] = \begin{cases} a_k + \sum_{k=1}^{n-1} \frac{k}{n} c[k] a_{n-k} & \text{if } 1 \leq n \leq p \\ \sum_{k=n-p}^{n-1} \frac{k}{n} c[k] a_{n-k} & \text{if } n > p. \end{cases} \tag{6}$$

The first 21 LPCs and 10 LPCCs (apart from the zeroth order) were included in our dataset. An equivalent measure to these that has become popular in speaker analysis is line spectral frequencies (LSFs) [34]. These can be useful in practice as they result in low spectral distortion and are deemed to be more sensitive and efficient than other equivalent representations. The first 20 LSFs were included in the dataset resulting in a total of 49 linear features.

Perceptual Linear Prediction One downfall of the LP method is that it approximates the spectrum of speech equally well at all frequencies. In contrast, after 800Hz, the human ear becomes less sensitive and spectral resolution decreases with frequency. This is compensated for using Perceptual Linear Prediction (PLP) [29]. This combines three concepts from the psychophysics of human hearing to improve the estimation of the auditory spectrum: the critical band spectral resolution, the equal-loudness curve and the intensity power law. The critical band spectral resolution warps the spectrum into the Bark frequency and then convolves it with the power spectrum of the simulated critical-band masking curve. The equal loudness curve is used to to pre-emphasise the resultant samples. It is then subjected to the cubic-root amplitude compression which approximates the power law of hearing. PLP and other short term spectral values are vulnerable when the values are modified by the frequency response of the channel. The RelAtive SpecTrAl (RASTA) [30] method was developed to make PLP more robust to linear spectral distortions by replacing the short-term spectrum by a spectral estimate. This suppresses any slow varying component making the spectral estimate of that channel less sensitive to slow variations and noise. PCA was again used to reduce the dimensionality of the first nine PLPs and RASTA PLPs resulting in 72 discrete measures for these features.

Other Features A number of descriptive spectral features were also included in our dataset. These features have been found to be very useful in musical instrument identification [52], but are not typically used for speech analysis. Ten specific values were included: the Spectral Centroid, Inharmonicity, Number of Spectral Peaks, Zero Crossing Rate, Spectral Rolloff, Brightness, Spectral Regularity and the Spectral Spread, Skewness and Kurtosis. Many of these were calculated using the MIRToolbox [45], a Matlab toolbox dedicated to the extraction of musically-related features from audio recordings.

3.5 Primitive language, variation operators, GP parameters

The primitive language and the evolutionary run parameters are given in Table 1. Many of these are typical values taken from the literature for problems of a similar size. We include 40 randomly chosen constants to create an approximate 7:1 ratio of features:constants. This creates a bias towards a program tree selecting a feature as a terminal node while allowing opportunity for it to pick a constant if necessary. Each generation, the ST and AVE methods perform 120,000 fitness evaluations, whereas the US and RS only perform 24,000 and 2,000 respectively. To ensure a fair comparison between experiments, it is imperative that the same the number of calculations are performed across their duration and so the number of generations used in the US and RS methods are adjusted accordingly. Preliminary experiments revealed a tendency of all systems to overfit, thus the maximum tree-depth is set to 8 to restrict the complexity of the evolved programs.

The search strategy that we employed relies heavily on mutation-based variation operators. The operation of `pointMutation(x)` traverses the tree in a depth-first manner, and depending on the probability x it substitutes a tree-node by another random tree-node of the same arity. The operation of `subtreeMutation()` selects a node uniform-randomly and replaces the subtree rooted at that node with a newly generated subtree. The tree-generation procedure is *grow* or *full*, each applied with equal probability. To improve on the exploratory effect of the mutation operator, other than picking the tree-node to be replaced from the whole expression-tree, we devised an additional node-selection method. In this method a depth-level is picked uniform-randomly from the range of all possible depth-levels present in the expression-tree, and subsequently a node is picked uniform-randomly from the set of nodes that lie in the chosen depth-level. The decision between the two node-selection methods is governed by a probability set to 0.5 for both methods. Finally, our implementation of recombination operator is the standard subtree crossover defined for expression-tree representations. The probability of selecting an inner-node

Table 1 Function/Terminal sets and run parameters

PRIMITIVE LANGUAGE	
Function set	+, −, ∗, / (x/y returns x if $\|y\| < 10^{-5}$), *sin*, *cos*, e^x, *log* (log(x) returns x if $x \leq 0$), *sqrt* (sqrt(x) returns x if $x < 0$)
Terminal set	275 features 40 uniform-randomly drawn constants in the range of $[-1.0, 1.0]$
GP PARAMETERS	
Evolutionary algorithm	elitist (1% of population size), generational
Population size	1,000
Tournament size	4
No. of generations	51 for ST 51 for AVE 251 for US 3,001 for RS
Population initialisation	ramped half-and-half (depths of 2 to 4)
Max. tree depth	8
Crossover Probabilities	inner node probability: 0.9 leaf node probability: 0.1
Mutation Probabilities	pointMutation(0.1): 0.1 pointMutation(0.2): 0.1 pointMutation(2/treesize): 0.2 subtree mutation: 0.6

as a crossover point is set to 0.9, while the probability of selecting a leaf-node is set to 0.1.

In generating offspring, a probability is associated with applying either mutation or crossover, set to 0.7 in favour of mutation. If mutation is chosen, `pointMutation(0.1)` is applied with a probability of 0.1, `pointMutation(0.2)` is applied with a probability of 0.1, `pointMutation(2 / tree_size)` is applied with a probability of 0.2, and `subtreeMutation()` is applied with a probability of 0.6.

3.6 GP for Feature Selection

As detailed above, 275 features were used for these original GP experiments. In ASV, as in many other classification problems, the features used are generally chosen from those found in the literature on alternative methods applied to the given problem; there is rarely experimental justification for the inclusion of specific features. One of the best practical aspects of GP is that it results in a white-box system, meaning that the the final trees can be observed. The implication of this is that successful trees can be analysed to determine which temporal and spectral features are included to make them successful. Two specific feature selection methods termed the *Original Selection* and *Accuracy Selection* methods were derived from our GP results as detailed below.

3.6.1 'Original' Method for Feature Selection

We can determine the most beneficial features for the given problem by examining which features are most often chosen by high performing trees. A similar method has been used for feature selection in musical instrument analysis [51,

```
tree = current_tree;
tree_acc = classification_accuracy(tree);
if (tree_acc > 90%) {
  for feature in tree {
    do 1000 times{
      shuffled_tree = shuffle(tree(feature)); %shuffle the feature value
      acc = classification_accuracy(shuffled_tree); %calculate the new accuracy
      }
    new_acc = average(acc)
    feature_acc = abs(new_acc-tree_acc)
    }
else {
  tree = next_tree; %move to the next program tree
}
```

Fig. 1 Outline of Accuracy Selection method

52]. In examining these successful features, we only consider those program trees from 50 independent GP runs that attain a test classification accuracy greater than 90%. This benchmark was chosen because in employing a 1:9 class imbalance, a naive system that merely classifies all data into one class would be able to achieve 90% accuracy. Hence we only consider programs that achieve higher than this to be successful. The number of times each feature was used in one of these successful trees is noted. The top 20 features as chosen by successful trees can then be selected for training and testing a number of machine learning classifiers. As this is based on the trees evolved from our original GP experiments we termed this selection method *Original Selection.*

3.6.2 'Accuracy' Method for Feature Selection

The second method of feature selection is based on the degree of change in classification accuracy that results from a change in a given feature. To distinguish this from the original count-based method of selecting features we term this method *Accuracy Selection.* A similar method of feature selection was used in studying the use of GP on the oral bioavailability problem in [19]. Again only trees that achieve over 90% test classification are considered successful and used for feature selection. For each feature in the successful tree, its value is shuffled among the values present in the test set and the test classification accuracy is recalculated. This process is repeated 1000 independent times and an average value is obtained. The absolute percentage change is calculated between the original classification accuracy and the average obtained by shuffling this feature. This process is outlined in Figure 1. A large absolute percentage change in accuracy for a given feature indicates that this feature is important. The top 20 important features obtained in this manner can then be selected to train a number of classifiers.

These two sets of selected features by each of the four GP methods were compared against the full set of 275 features used to train and test four Machine Learning Classifiers, namely Support Vector Machines (SVM), Logistic Regression (LR), Random Forest (RF) and Gradient Boosting Classifier (GBC).

4 Results

This section reports results on the generalisation performance of the GP systems, comparison of the GP system with other tuned Machine Learning methods, using GP for feature selection on these other methods and a comparison between the speakers.

For each experiment, we created 50 splits of the 200 learning examples into training and test sets. In each split, 120 examples were drawn uniform-randomly for the training set, while the remaining 80 examples populate the test set. Stratification ensures that the class imbalance ratio is maintained in both sets. Using each split, we performed 50 independent evolutionary runs using each GP system for each of the to-be-verified speakers. Many practitioners use an equal weighing in the AVE system by setting $w = 0.5$ [7]. In this work the effectiveness of AVE is evaluated using a set of values for w, that of $W = \{0.5, 0.6, 0.7, 0.8\}$. In the experiments we performed no model selection, thus the fittest individual (on the training dataset) of the last generation is designated as the output of a run.

4.1 Generalisation performance

Table 2 presents statistics of training and test *classification accuracy* for the different systems on all 10 speakers. In each case, we report the median, interquartile range, and maximum based on 50 independent runs. Note that in a classification setup, in which the true positive rate corresponds to the minority class accuracy, a classifier that always outputs the majority class label attains a classification accuracy of 0.9 (true positive rate of 0%). Our first observation concerns the significant difference between training and test performance in all datasets. This is indicative of overfitting, a typical problem in unregularised GP [1]. There are a number of reasons why overfitting is occurring in these preliminary experiments. First and foremost, this is attributed to the limited number of examples for the to-be-verified speakers in each dataset. A second reason is the absence of both model selection and regularisation from the learning process. In light of the above, we attempted to limit the syntactic complexity of the evolved programs by setting the maximum tree-depth allowed during search to 8, however this was not adequate for preventing overfitting.

The generalisation performance of different systems is presented in the second part of Table 2. Table 4 presents the p-values of a two-sided Wilcoxon rank sum test, which tests the *null* hypothesis that two data samples have equal medians, against the alternative that they don't. We set the significance level α to 0.05. Median test accuracy of ST is shown to be statistically superior to rest of the systems AVE, RS, US for speakers FJEN0, MMGC0, MPGR1. In addition, ST median is shown to be statistically superior against that of (a) AVE for speakers FPJF0, FSAH0; (b) RS for speakers FGRW0, FPJF0, MTRT0; and (c) US for speakers MJDC0. This result is consistent with the findings in [7], which showed that the MSE-based loss function of Equation 1 routinely outperformed loss functions based on classification accuracy or the weighted average between true positive and true negative rates (Equation 3). The results also suggest that ST, which uses the original unbalanced datasets, is often statistically superior or no different to the data-sampling methods of RS and US. Specifically ST is statistically better in 6/10 speakers, and statistically worse in 1/10 speakers against RS. Also, ST is statistically better in 4/10 speakers, and statistically worse in 1/10 speakers against US.

Overall, the median of ST is equal to 90% in 6/10 speakers, and greater than 90% in 4/10 speakers. This suggests that in 50 runs, the median generalisation performance of ST is consistently equal or better to the performance of a classifier that always outputs the majority class label. The median test accuracy is higher than or equal to 90% in 4/10 speakers for AVE; 4/10 speakers for RS; and 5/10 speakers for US. Nevertheless, among 50 runs, the maximum test classification accuracy that is achieved through evolution is always higher than the one yielded from the classifier that always outputs the majority class label, for all speakers.

Table 3 presents the test accuracy statistics for the different values of w in the loss function (Equation 3) of the Ave system. A two-sided Wilcoxon rank sum test is performed to test the difference in the median values. We found that no value of w, where $w \neq 0.5$ resulted in a significantly better test classification accuracy compared to equal weighing. This finding is in accordance with the result reported in [7].

4.2 Comparison with Other Classifiers

To evaluate the performance of the GP classification systems we compared the results from the best generalising system (ST) against a number of tuned machine learning classifiers, namely a Support Vector Machine (SVM), Random Forest (RF), Logistic Regression (LR) and Gradient Boosting Classifier (GBC). A parameter search was performed to optimise each classification model from the parameters shown in Table 5.

The Training and Testing classification accuracies for GP and the four machine learning methods are shown in Figure 2. From this figure we can again see a very noticeable reduction in accuracy between train and test performance for all methods. While the training classification accuracy for a number of methods on a number of speakers is close to 100%, it is clear that this classification accuracy rarely exceeds the majority threshold of 90% in any of the test classifications. This shows that it is not only GP that suffers from over-fitting in classification problems like this that contain such a severe class imbalance. In regards to test classification accuracy, of the 10 speakers, SVM performed

Table 2 Training and Test performance for all systems showing the median (with interquartile range in parenthesis) and maximum accuracy for each speaker based on 50 independent runs.

Training Classification Accuracy

	AVE($w = 0.5$)		**ST**		**RS**		**US**	
Speaker	**Median**	**Max**	**Median**	**Max**	**Median**	**Max**	**Median**	**Max**
FGRW0	0.97 (0.04)	1.00	0.98 (0.09)	1.00	0.96 (0.03)	0.99	0.99 (0.03)	1.00
FJEN0	0.97 (0.05)	1.00	0.95 (0.09)	1.00	0.97 (0.02)	0.98	0.98 (0.03)	1.00
FPJF0	0.99 (0.03)	1.00	0.93 (0.08)	1.00	0.97 (0.03)	1.00	0.98 (0.03)	1.00
FSAH0	0.97 (0.05)	1.00	0.90 (0.06)	0.98	0.97 (0.03)	0.99	0.98 (0.03)	1.00
MEFG0	1.00 (0.02)	1.00	0.98 (0.07)	1.00	0.98 (0.02)	1.00	0.99 (0.01)	1.00
MJDC0	0.97 (0.04)	1.00	0.97 (0.07)	1.00	0.96 (0.01)	0.98	0.99 (0.03)	1.00
MKDD0	0.98 (0.04)	1.00	0.97 (0.09)	1.00	0.97 (0.03)	1.00	0.99 (0.02)	1.00
MMGC0	0.95 (0.04)	1.00	0.92 (0.07)	0.99	0.96 (0.03)	0.98	0.97 (0.03)	1.00
MPGR1	0.97 (0.03)	1.00	0.97 (0.06)	1.00	0.96 (0.03)	0.98	0.98 (0.04)	1.00
MTRT0	0.96 (0.03)	1.00	0.90 (0.05)	0.98	0.96 (0.03)	0.99	0.97 (0.03)	1.00

Testing Classification Accuracy

	AVE($w = 0.5$)		**ST**		**RS**		**US**	
Speaker	**Median**	**Max**	**Median**	**Max**	**Median**	**Max**	**Median**	**Max**
FGRW0	0.90 (0.06)	0.97	0.91 (0.03)	0.96	0.90 (0.04)	0.97	0.93 (0.04)	0.97
FJEN0	0.85 (0.06)	0.91	0.91 (0.01)	0.95	0.89 (0.03)	0.96	0.88 (0.02)	0.96
FPJF0	0.86 (0.05)	0.95	0.90 (0.04)	0.95	0.88 (0.04)	0.94	0.89 (0.10)	0.96
FSAH0	0.84 (0.05)	0.91	0.90 (0.01)	0.95	0.89 (0.04)	0.96	0.90 (0.05)	0.95
MEFG0	0.93 (0.05)	0.99	0.90 (0.05)	0.99	0.94 (0.03)	0.97	0.91 (0.05)	0.96
MJDC0	0.90 (0.08)	0.96	0.90 (0.04)	0.96	0.90 (0.04)	0.94	0.88 (0.02)	0.94
MKDD0	0.93 (0.07)	1.00	0.90 (0.04)	0.95	0.91 (0.05)	0.97	0.94 (0.04)	1.00
MMGC0	0.81 (0.14)	0.94	0.91 (0.01)	0.94	0.86 (0.04)	0.93	0.86 (0.06)	0.94
MPGR1	0.79 (0.10)	0.94	0.92 (0.03)	0.93	0.88 (0.06)	0.94	0.85 (0.05)	0.93
MTRT0	0.89 (0.07)	0.94	0.90 (0.01)	0.94	0.88 (0.05)	0.94	0.90 (0.05)	0.95

Table 3 Test classification accuracy for AVE. Interquartile range in parentheses.

	AVE($w = 0.5$)		**AVE**($w = 0.6$)		**AVE**($w = 0.7$)		**AVE**($w = 0.8$)	
Speaker	**Median**	**Max**	**Median**	**Max**	**Median**	**Max**	**Median**	**Max**
FGRW0	0.90 (0.06)	0.97	0.88 (0.11)	0.99	0.84 (0.10)	0.97	0.86 (0.14)	0.96
FJEN0	0.85 (0.06)	0.91	0.85 (0.06)	0.93	0.82 (0.10)	0.90	0.82 (0.10)	0.93
FPJF0	0.86 (0.05)	0.95	0.89 (0.12)	0.95	0.86 (0.09)	0.95	0.86 (0.06)	0.94
FSAH0	0.84 (0.05)	0.91	0.86 (0.06)	0.94	0.82 (0.10)	0.96	0.84 (0.06)	0.91
MEFG0	0.93 (0.05)	0.99	0.93 (0.06)	1.00	0.91 (0.06)	0.95	0.94 (0.05)	0.97
MJDC0	0.90 (0.08)	0.96	0.86 (0.06)	0.96	0.78 (0.12)	0.90	0.84 (0.08)	0.95
MKDD0	0.93 (0.07)	1.00	0.88 (0.12)	0.97	0.91 (0.10)	0.99	0.88 (0.12)	0.99
MMGC0	0.81 (0.14)	0.94	0.81 (0.08)	0.90	0.81 (0.09)	0.90	0.84 (0.07)	0.94
MPGR1	0.79 (0.10)	0.94	0.79 (0.06)	0.93	0.85 (0.09)	0.93	0.82 (0.10)	0.94
MTRT0	0.89 (0.07)	0.94	0.85 (0.10)	0.94	0.85 (0.07)	0.96	0.84 (0.12)	0.95

the highest for three speakers, GP and RF for two and GBC for one. RF and GBC also performed equally highest for two speakers. Hence, no one method performs consistently higher across all speakers.

To statistically confirm this overall performance, we conducted a Friedman aligned ranking test [31] across all methods for all speakers to test the *null* hypothesis that there is no significant performance difference between the classifiers across all speakers. This resulted in a p-value of 0.28 indicating that we must accept the hypothesis that there is not a signifiant performance dif-

Table 4 p-values of Winlcoxon rank-sum test. AVE uses $w = 0.5$.

		ST	RS	US			ST	RS	US
FGRW0	AVE	0.27	0.50	0.03	**FJEN0**	AVE	0.00	0.00	0.00
	ST		0.03	0.08		ST		0.04	0.00
	RS			0.00		RS			0.09
		ST	RS	US			ST	RS	US
FSAH0	AVE	0.00	0.00	0.00	**MEFG0**	AVE	0.00	0.42	0.01
	ST		0.17	0.47		ST		0.00	0.65
	RS			0.16		RS			0.00
		ST	RS	US			ST	RS	US
MKDD0	AVE	0.35	0.71	0.00	**MMGC0**	AVE	0.00	0.00	0.00
	ST		0.12	0.00		ST		0.00	0.00
	RS			0.00		RS			0.87
		ST	RS	US			ST	RS	US
MTRT0	AVE	0.21	0.28	0.19	**FPJF0**	AVE	0.02	0.32	0.20
	ST		0.00	0.80		ST		0.02	0.92
	RS			0.00		RS			0.23
		ST	RS	US			ST	RS	US
MJDC0	AVE	0.17	0.79	0.05	**MPGR1**	AVE	0.00	0.00	0.00
	ST		0.03	0.00		ST		0.01	0.00
	RS			0.00		RS			0.00

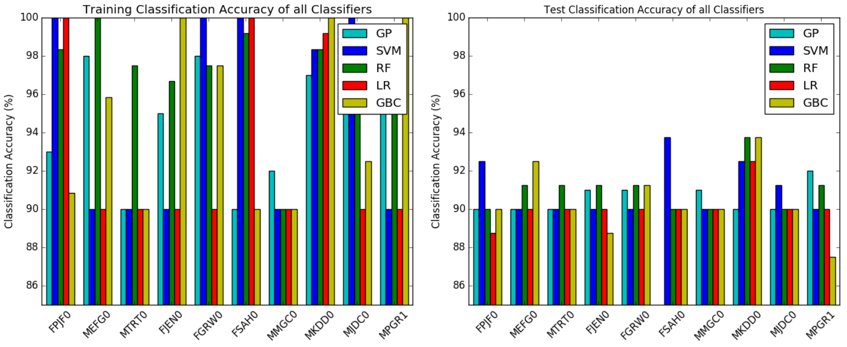
Table 5 Parameters for optimisation for each of the machine learning classifiers

SVM	C: [0.01, 0.1, 1, 10, 100] Gamma: [0.1, 0.01, 0.001, 0.0001] kernel: [‘rbf’]
RF	no. estimators: [10, 50, 100, 200, 300, 400, 500, 1000] max depth: [1,2,3, None] max features: [1, 3, 5, 10, 50, 100] min samples split: [1, 5, 10, 20] min samples leaf: [1, 5, 10, 20] bootstrap: [True] criterion: [‘gini’, ‘entropy’]
LR	C: [0.0001, 0.001, 0.01, 0.1, 1]
GBC	no. estimators: [10, 50, 100, 200, 300, 400, 500, 1000] learning rate: [0.001, 0.01, 0.1] max depth: [1,2,3]

ference between all classifiers across all speakers. In the next section we try to improve this result by optimising each classifier using only specifically chosen features from the two feature selection methods described in Section 3.6.

4.3 Feature selection with GP

As described in Section 3.6, the white-box nature of GP facilitates an analysis of the evolved tree programs. Using GP to create classifiers with a wide range of possible features allows us to determine the most beneficial features for the given problem by examining which are most often selected by high performing systems. Thus we can use the experiments above to determine which may be the best features to include for use with other machine learning classifiers. To do this, we only analysed successful classifiers, those that achieved over 90% test classification accuracy. We compare the two methods of feature selection below.



(a) (b)

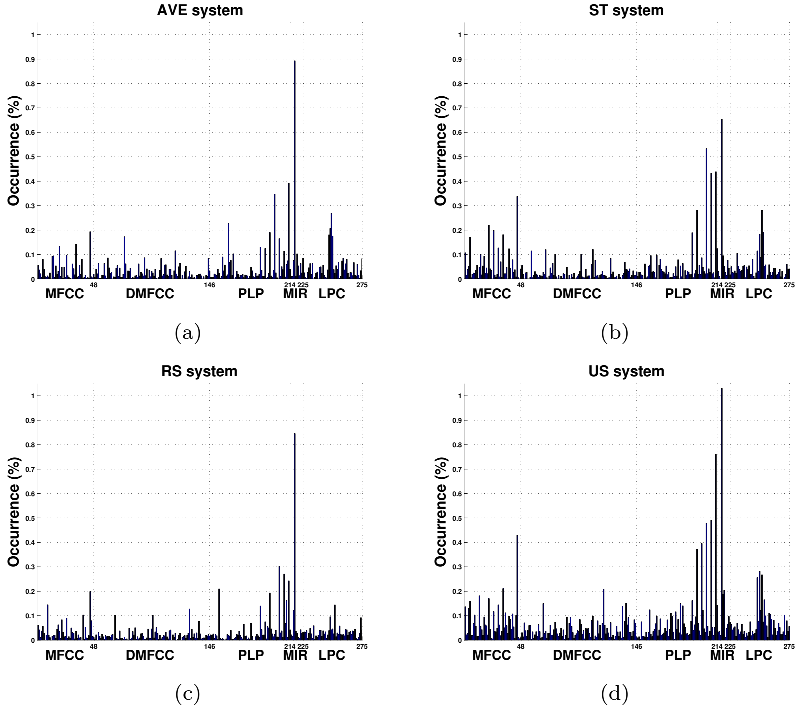
Fig. 2 Training and Testing Classification Accuracy for all classifiers using all data

4.3.1 Original Selection

A plot of the mean percentage of times each feature is chosen by a successful classifier is shown in Figure 3. A more detailed account of the top 20 chosen features for each system is given in Table 6. This overall set of 20 most popular chosen features from each GP system was selected to train and test the next set of machine learning classifiers. This table names each of the top chosen features, reporting the mean percentage of times this feature was chosen from the list of all 275 features and the standard error of this selection.

From the plots in Figure 3 it is clear that certain features are chosen more consistently by high performing classifiers than others. In each system investigated there is a strong peak at feature number 218. We can see from Table 6 that this corresponds to Inharmonicity. If a sound is perfectly internally 'harmonious' each of the upper partials will be integer multiples of the fundamental frequency. Inharmonicity is a measure of how much the spectral content of a sound differs from this ideal relationship. Although it has been generally used as a musical descriptor, its prominent and consistent selection in high performing classifiers in these experiments indicate that it may be a very strong indicator for voice verification also. Other individual spectral features are not strongly represented although the Zero Crossing Rate, Spectral Centroid and the Number of Spectral Peaks did appear in the top 20 features chosen by at least one system.

From Table 6 we can see that higher order PLPs were the next most selected feature. Within these only the first PC was chosen, indicating that the variance in the principle dimension for these features contains the most useful information. Surprisingly, the RASTA variations were not selected as frequently implying that the original implementation of the PLPs are more important for this problem. This is most likely because we used the high quality audio signal from the TIMIT database without adding noise. The RASTA method was developed to compensate for noisy channels, but as our signals



(a)

(b)

(c)

(d)

Fig. 3 Mean of the occurrence of all features in the top performing programs from independent runs for the AVE, ST, RS and US systems described in Section 3.3. Features are grouped into the mel-frequency cepstral coefficients (MFCC), their derivatives (DMFCC), perceptual linear prediction (PLP), other spectral features (MIR) and the linear prediction coefficients (LPC).

are not noisy these are not found to be more beneficial than the standard PLP implementation.

The LPCCs were prominent among the highly selected features. LPCC3 was within the top 20 for each system and LPCCs 7 and 5 also featured in three of the four systems. Interestingly, the LPCs did not feature as strongly as their cepstral counterparts, indicating that in linear prediction for these problems the cepstral domain may be more influential than the spectral domain. MFCCs have for a long time been one of the most widely used features in speech analysis. It may be surprising then to see that they did not appear as prominently as other features already discussed. In saying that, the first PC of a number of higher MFCCs did emerge as consistently chosen by successful classifiers. The derivatives of the MFCCs were among the least successful features.

These results show a clear tendency towards the selection of certain spectral features over others. Such a result shows how important it is to consider feature selection when designing machine learning classification experiments.

Table 6 Top 20 features Originally selected by each system. Values are reported as the mean percentage of times each feature was chosen by a successful classifier (accuracy greater than 90%) with the standard error in parenthesis.

Ave		ST		RS		US	
Feature	**Mean(%)**	**Feature**	**Mean(%)**	**Feature**	**Mean(%)**	**Feature**	**Mean(%)**
Inharm	0.89 (0.38)	Inharm	0.66 (0.2)	Inharm	0.85 (0.16)	Inharm	1.03 (0.23)
plp9_1	0.39 (0.18)	plp7_1	0.53 (0.1)	plp7_1	0.3 (0.11)	plp9_1	0.76 (0.15)
plp6_1	0.34 (0.27)	plp9_1	0.44 (0.11)	plp8_1	0.27 (0.06)	plp8_1	0.49 (0.18)
lpcc4	0.27 (0.1)	plp8_1	0.43 (0.11)	plp9_1	0.24 (0.06)	plp7_1	0.47 (0.09)
plpR5_2	0.23 (0.13)	mfcc12_1	0.34 (0.08)	plpR3_2	0.21 (0.01)	mfcc12_1	0.43 (0.1)
lpcc3	0.2 (0.15)	lpcc7	0.28 (0.09)	mfcc12_1	0.19 (0.03)	plp6_1	0.4 (0.16)
mfcc12_1	0.19 (0.04)	plp5_1	0.28 (0.1)	plp5_1	0.19 (0.05)	plp5_1	0.37 (0.1)
plp5_1	0.19 (0.1)	mfcc6_1	0.22 (0.1)	plp8_3	0.16 (0.05)	lpcc5	0.28 (0.14)
lpcc2	0.18 (0.17)	mfcc7_1	0.2 (0.08)	mfcc3_1	0.14 (0.04)	lpcc7	0.27 (0.16)
lpcc5	0.18 (0.1)	lpcc8	0.19 (0.09)	lpcc7	0.14 (0.03)	lpcc3	0.26 (0.1)
Dm7_2	0.17 (0.07)	plp4_1	0.19 (0.07)	plp3_1	0.14 (0.09)	mfcc9_1	0.21 (0.06)
plp7_1	0.17 (0.06)	lpcc5	0.18 (0.09)	DDm9_1	0.13 (0.02)	DDm6_2	0.21 (0.06)
mfcc9_1	0.14 (0.07)	mfcc9_1	0.18 (0.06)	Centroid	0.12 (0.02)	ZeroC	0.2 (0.09)
mfcc5_3	0.14 (0.09)	mfcc2_1	0.17 (0.08)	mfcc10_3	0.1 (0.00)	NoPeaks	0.19 (0.1)
plp3_1	0.13 (0.12)	mfcc8_1	0.12 (0.06)	Dm5_2	0.1 (0.00)	mfcc4_1	0.18 (0.05)
plp4_1	0.12 (0.08)	plp9_2	0.12 (0.05)	DDm1_2	0.1 (0.00)	mfcc6_1	0.17 (0.07)
DDm6_1	0.12 (0.11)	mfcc10_2	0.12 (0.08)	lpcc3	0.1 (0.04)	lpcc9	0.17 (0.06)
plp8_1	0.11 (0.08)	DDm4_1	0.12 (0.05)	lsf19	0.09 (0.09)	plp4_1	0.16 (0.04)
plpR6_2	0.1 (0.05)	Dm6_1	0.12 (0.07)	mfcc7_1	0.09 (0.04)	mfcc2_1	0.16 (0.11)
ZeroC	0.1 (0.1)	lpcc3	0.11 (0.06)	mfcc6_1	0.08 (0.05)	DDm11_1	0.15 (0.05)

Evolutionary computational techniques have been shown to be useful for this kind of feature selection [65]. Hence we have used the top 20 features as chosen by this Original Selection to try to optimise the performance of the machine learning classifiers.

4.3.2 Accuracy Selection

The second Accuracy Selection method for feature selection considers the change in accuracy in each speaker when the value of each feature used for that speaker is altered. In this case, the top 20 features for each individual speaker were chosen to train and test the next set of machine learning classifiers for that speaker. This Accuracy Selection method was used to test if selecting features deemed important to the specific speaker would increase generalisation.

A comparison of the test classification accuracies on each of the methods for both feature selection methods is shown in Figure 4. These would appear to indicate that features included from the Original Selection method increased generalisation for more methods than those found from the Accuracy Selection. To examine this more closely, Table 7 shows the mean and standard error of the classification across all speakers for each of the methods. This shows a improvement in the mean classification accuracy when Original Selection is employed over no feature selection, despite a reduction in data from 275 to only 20 selected features. This clearly indicates that many of the included features (chosen from those commonly used in the literature) were superfluous or even

Table 7 Test classification accuracy for each ML classifier for each GP system averaged across all ten speakers for both Original and Accuracy feature selection methods. The *none* values report classification accuracy for each classifier without any Feature Selection.

		Original FS		Accuracy FS	
ML	**GP**	**Mean**	**Std. Dev**	**Mean**	**Std. Dev**
SVM	none	91.0	1.35	91.0	1.35
	AVE	92.75	2.78	90.25	1.66
	ST	93.25	2.51	91.13	1.8
	US	92.5	3.0	90.88	2.4
	RS	91.0	1.56	92.75	2.7
RF	none	91.0	1.1	91.0	1.1
	AVE	92.63	1.72	90.63	1.88
	ST	92.38	2.34	90.75	2.18
	US	92.13	2.1	90.25	3.3
	RS	92.5	1.37	91.75	1.6
LR	none	90.13	0.88	90.13	0.88
	AVE	90.13	0.38	90	0
	ST	90.13	0.38	90	0
	US	90.13	0.38	90	0
	RS	90	0	90	0
GBC	none	90.37	1.68	90.37	1.68
	AVE	91.75	1.39	90.13	3.13
	ST	91.75	2.18	90.75	2.69
	US	92.25	1.56	90	3.6
	RS	92.63	1.78	94.38	1.79

detrimental to the system. The results obtained using the Original Selection method outperform those from the Accuracy method for each machine learning classifier from features selected by all but the RS GP systems. This would indicate that taking the instances of a given feature in a successfully evolved GP classifier is a better method of feature selection than considering the change in accuracy resulting from a change in that feature.

Figure 4 also shows a strong difference in performance of the Machine Learning Classifiers. Most notably, regardless of the GP system or feature selection used, the LR very rarely achieves a classification accuracy of higher than the benchmark of 90%. This result is also shown in Table 7 whereby, regardles of GP method used, neither feature selection method manages to improve the original classification that used the full set of 275 features. This would imply that this method is not good at generalising for highly unbalanced classification studies such as this. Each of the other methods suffer from stagnation at 90% accuracy for a number of runs, but each of the SVM, RF and GBC methods manage to improve this to over 95% for a least one run in the results shown in Figure 4.

Table 8 p-values for aligned Friedman tests for each Classifier with respect to GP methods using Original Selection

	AVE	ST	US	RS
p-value	0.188	0.916	0.897	0.694

Table 9 p-values for aligned Friedman tests for each GP method with respect to Classifier using Original Selection

	SVM	RF	LR	GBC
p-value	0.005	0.008	0.029	0.0007

4.3.3 Analysis

To confirm the significance of these results we ran a number of statistical analyses. To consider the effect of the GP method on each classifier we first ran a Friedman aligned ranking test [31] across each of the four GP methods (AVE, ST, US and RS) for each of the classifiers across all speakers. The p-values from this analysis using the Original Selection method is shown in Table 8. If we consider a significance level of 0.05, it is clear from the high value of each of these results that we can accept the null hypothesis that there is no significant performance difference on any classifier depending on the GP method chosen. Table 9 shows the results of another Friedman aligned ranking test in respect to the different classifiers used with each GP method. In this case the small (< 0.05) obtained p-values mean that we must reject the null hypothesis that there is no significant performance difference in choosing the classification method (i.e. SVM, RF, LR and GBC). To examine this further we conducted a post-hoc analysis using the Holm-Bonferroni method [32]. This sequential post-hoc correction performs pair-wise analyses to control the family-wise error rate among multiple comparisons. The results in Table 10 indicate that it is particularly the performance of the LR method that causes a significant change in performance, which agrees with Figure 4 showing the LR classifier to have notably poor performance across a number of speakers.

The results of a similar Friedman test for the Accuracy Feature Selection with respect to the chosen GP method and chosen ML classifier are shown in Tables 11 and 12 respectively. Considering a significance level again of 0.05, we do not see a significant difference in the performance of the variations of the system and determine that equivalent null hypotheses could be accepted in all but the GBC and RS variations of the experiment.

Figure 4 shows a marked increase in accuracy in a number of experiments but we would like to clarify how many times applying feature selection resulted in an improvement in test classification accuracy. To do this, we counted the number of speakers for which each feature selection method improved test classification accuracy in comparison to the classification results using all features shown in Figure 2. The results are tabulated in Table 13. This clearly shows that the application of feature selection results in an increase in classification

Table 10 p-values of post-hoc analysis using the Holm-Bonferroni method

		RF	LR	GBC
AVE	SVM	0.81	0.003	0.81
	RF		0.0007	0.42
	LR			0.01
		RF	LR	GBC
ST	SVM	0.38	0.0007	0.166
	RF		0.015	0.768
	LR			0.05
		RF	LR	GBC
US	SVM	1	0.02	1
	RF		0.02	1
	LR			0.01
		RF	LR	GBC
RS	SVM	0.03	0.15	0.058
	RF		0.0002	0.69
	LR			0.0006

Table 11 p-values for aligned Friedman tests for each Classifier with respect to GP methods using Accuracy Selection

	AVE	ST	US	RS
p-value	0.09	0.25	1.0	0.005

Table 12 p-values for aligned Friedman tests for each GP method with respect to Classifier using Accuracy Selection

	SVM	RF	LR	GBC
p-value	0.78	0.49	0.75	0.001

Table 13 Number of speakers (out of 10) for which each method of feature selection increased (⇑), had no effect (⇔) or decreased (⇓) the test classification on each of the machine learning classifiers.

		Original FS			Accuracy FS		
		⇑	⇔	⇓	⇑	⇔	⇓
SVM	Ave	7	2	1	1	4	5
	ST	8	1	1	3	5	2
	US	5	4	1	2	4	4
	RS	2	6	2	6	3	1
RF	Ave	7	1	2	3	2	5
	ST	6	2	2	4	2	4
	US	4	3	3	3	2	5
	RS	8	1	1	5	5	0
LR	Ave	1	8	1	1	8	1
	ST	2	7	1	1	8	1
	US	1	8	1	1	8	1
	RS	1	8	1	1	8	1
GBC	Ave	5	4	1	3	5	2
	ST	4	6	0	6	2	2
	US	6	3	1	2	3	5
	RS	7	3	0	10	0	0

more often than a decrease. Again the Original Selection method performs stronger than the Accuracy Selection method. Specifically, the application of Original Selection resulted in an increase of classification accuracy in a total of 74 individual experiments and a decrease in only 19; using Accuracy Selection resulted in an increase of 52 and a decrease in 39. This means that in reducing 275 features to 20 using the Original Selection we obtained test classification results that are as good as or better than those obtained using all data in 88.12% of experiments undertaken (75.6% using Accuracy Selection). Notably, the test classification results for the LR methods showed little increase or decrease in accuracies — the accuracy of most speakers were not changed regardless of feature selection.

4.4 Comparison Between Speakers

Regardless of the method used above, it is clear from the results that classification models created for certain speakers are more successful than for others. To examine this we once again consider the results shown in Figure 4, but this time in regards to the individual speakers. We again used a Friedman aligned ranking test across our results to determine if there is a significant difference in performance between speakers. We obtained a p-value of 0.0001 for the Original Selection and 0.0007 for Accuracy Selection determining that there is a significant difference between the performance of the speakers. Table 14 shows the maximum, average and standard deviation for each of the speakers over all systems using both feature selection methods. We can see from this table that the maximum achieved accuracy for both feature selection methods was for speaker FGRW0 at 98.75% for both models. From Figure 4 we can see that this result was achieved by an SVM in both instances, from the AVE system using the Original Selection and the US system using the Accuracy Selection. This speaker does not have the highest average accuracy however, this is MKDD0 and MEFG0, both of whom also achieve a high maximum accuracy also. The most difficult speakers to verify are FJEN0 for the Original Selection, achieving an average accuracy of 90.23% and only reaching a maximum of 91.25%. This speaker also appeared to be difficult for systems using the Accuracy Selection, achieving an average and maximum classification accuracy of 89.92% and 93.75% respectively.

The difference in accuracies between speakers is indicative of difficulties in the data rather than the ability of any of the proposed methods' ability to be able to overcome these difficulties. Throughout this study we have created classification models specifically for each individual speaker; each speaker proposes a different classification problem and some of those problems have proven easier to solve than others. We would like to concentrate further efforts in this domain on considering the differences in the problem posed by looking at different speakers.

Table 14 Average test classification accuracy for each speaker across all methods.

	Original FS			Accuracy FS		
Speaker	**Max**	**Mean**	**Std. Dev**	**Max**	**Mean**	**Std. Dev**
FPJF0	97.5	92.19	2.05	95.0	90.55	2.38
MEFG0	96.25	92.34	2.02	97.5	**92.89**	2.45
MTRT0	92.5	90.70	0.99	93.75	90.39	1.86
FJEN0	91.25	90.23	0.66	93.75	89.92	1.79
FGRW0	**98.75**	92.34	2.42	**98.75**	91.17	2.63
FSAH0	96.25	91.48	2.04	93.75	91.02	1.61
MMGC0	95.0	91.02	1.67	91.25	89.92	1.12
MKDD0	96.25	**92.97**	2.46	97.5	92.03	3.48
MJDC0	97.5	92.73	2.66	96.25	89.77	1.94
MPGR1	95.0	91.41	1.70	93.75	90.86	2.53

4.5 Comparison with Other Methods

The breadth of studies discussed in Section 2.1 is a testament to the work which is being undertaken in the field of ASV. Accurate and reliable ASV has important practical real-world applications and as such have received much attention from numerous focus groups in the academic community. State of the art methods are compared generally during the bi-annual NIST Speaker Recognition Evaluation which evaluates novel speaker recognition systems on a corpora of i-vectors based on phone recordings. As discussed in Section 2.1, this contest has become so popular as to create collaborations across multiple institutes and countries in an attempt to succeed in this challenge. The presented work does not evaluate on any of the proposed NIST corpora of i-vectors, but rather on real audio signals drawn from the TIMIT speaker database. The focus of this work has been on manipulating this raw data through data reduction (from feature selection) and in considering a more extreme class imbalance as has been considered in classification problems in alternative problem domains. While there has been limited study of evolutionary methods on ASV, we have seen many studies of evolutionary computation applied to unbalanced classification problems [8] and feature selection techniques [65]. We have not yet implemented one of the typical speaker representation such as i-vectors or noisy channel dependencies that have become popular in recent years, but instead framed the problem in a new way. Although this makes a direct comparison with state-of-the-art techniques impossible, we feel that this will add to the ever growing body of work focussed on ASV and can help in maintaining diversity within the field.

The TIMIT corpora used throughout this study has however been widely used in speech and speaker analysis for many years. Of the previous studies, the most similar experimental setup to the proposed study is in [15], however as discussed earlier, even this study differed in class imbalance (1:3 as opposed to 1:9), signal representation (noisy as opposed to noiseless), features used, results reported and the overall purpose of the experiment. For these reasons, a direct comparison of our results to other studies would not appear to be feasible or meaningful. Nevertheless, as a qualitative comparison, we briefly present some of the results from previous speech and speaker classification studies undertaken on the TIMIT corpora. [15] reported classification accuracies of between 79.6% to 97% for clean to various levels of noisy data. This improvement in noisy environments is a very successful result from this study. In a separate study looking at Superfeatures [16], depending on the model used, the best classification between music, noise or TIMIT speech, ranged from 76% to 100%. In [42] methods based on multiple time-domain windows (tapers) with frequency domain extraction were employed in MFCC extraction to develop robust ASV methods using both TIMIT and NIST corpora. Although the TIMIT corpora is used in the development of the system, evaluation is performed on two separate NIST databases. Multi-taper short term windows were again used in [55] on the TIMIT corpora in examining Gammatone filters for robust speaker verification. They present their results as equal error rates (EER), rather than classification error determining that using a Gammatone filter can be more effective than traditional triangular Mel filterbanks over multiple windows for ASV. [36] explored the use of non-negative matrix factorization for speaker recognition using the TIMIT corpus and reported classification results among eight speakers of up to 98.99% when a decision was made from majority voting based on the previous 1 second window and a maximum of 76% without any such voting. [10] used the Ntimit database, which is composed of clean speech signals from the TIMIT database recorded over local and long-distance telephone loops. They proposed an evolutionary strategy that optimised feature extraction complementarity of two speaker verification systems but again only in examining noisy signals. They report that their evolutionary strategy resulted in improvements (measured as EER) for both databases.

5 Conclusions and Future Work

This paper presented a study on the application of GP for feature selection on a highly unbalanced implementation of the binary classification problem of ASV. Initial experiments implemented four specific GP systems. An analysis of high performing systems facilitated the development of two separate feature selection methods termed Original Selection and Accuracy Selection. These feature selection algorithms were used in an attempt to optimise a series of machine learning algorithms, a SVM, RF, LR and GBC. The focus of the study was to highlight two real world considerations in dealing with ASV: data reduction through feature selection and severe class imbalance compensation.

In the first series of GP experiments, using a number of independent GP runs, it was possible to evolve good-generalising programs, but this generalisation was not consistent in terms of median performance across all runs for the majority of systems. The MSE-based loss function that measured the discrepancy between program output and target value attained a median generalisation performance that was at least as good as the 'majority classifier' for all speakers. This outperformed the loss function based on the weighted accuracy between minority and majority classes for most speakers. In addition, the MSE-based loss function performed better when used on the original unbalanced dataset than when used in combination with down-sampling in nearly all speakers. The use of non-equal weighted misclassification costs for the minority and majority classes did not significantly improve generalisation compared to an equal weighting.

This MSE-based GP method was found to have comparable classification results to other machine learning algorithms. Of the algorithms considered, Logistic Regression performed consistently weaker than the rest, rarely able to classify unseen samples better than a majority classifier. By examining high performing program trees evolved with GP, two feature selection methods were used to reduce the number of features from 275 to 20. Using feature selection noticeably improved the generalisation for a number of individual models. Specifically it was found that the Original Selection method formed by counting the number of instances of particular features in high-performing trees was very beneficial to a number of methods. In reducing 275 features to 20 using this Original Selection, we obtained test classification results that are as good as or better than those obtained using all data in 88.12% of experiments undertaken. These results indicate that a 1:9 class imbalance ratio poses a very difficult classification problem for any machine learning algorithm and that it is highly important to consider what data (i.e. features) to include when designing such experiments.

We noted that certain speakers are significantly easier to verify than others throughout the experiments. This is because each speaker poses a unique classification problem. In future work we would like to determine what makes one speaker more difficult to verify than another. By examining different speakers, and by analysing the features selected by methods such as those discussed, we hope to be able to investigate specific timbral qualities within the human speaking voice that are important for accurate verification.

As a classification problem, the class imbalance ratio of 1:9 is very high, but speaker verification poses a much higher imbalance in practice. True ASV requires the verification of one speaker from every other person in the world. We have considered scaling up the problem to a higher degree of imbalance but in reality, trying to verify one in a million amounts to detecting an anomaly in a dataset. Hence in future work, we may reframe and examine the problem again, not with a heavy class imbalance but by trying to verify one speaker against the rest of the corpus by implementing anomaly detection.

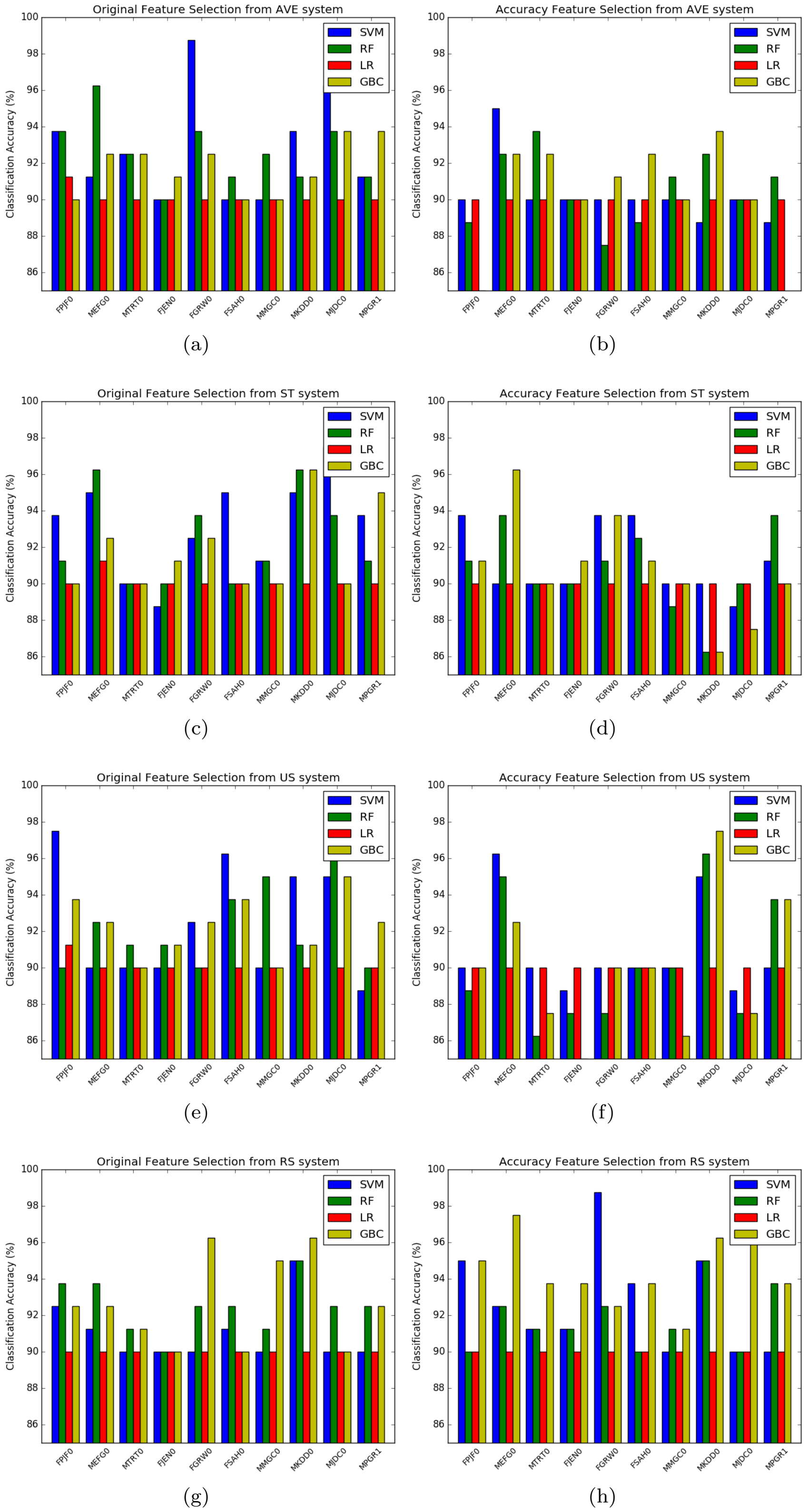
References

1. Agapitos, A., Brabazon, A., O'Neill, M.: Controlling overfitting in symbolic regression based on a bias/variance error decomposition. In: PPSN XII (part 1), *LNCS*, vol. 7491, pp. 438–447. Springer, Taormina, Italy (2012). DOI doi:10.1007/978-3-642-32937-1_44

2. Alegre, F., Amehraye, A., Evans, N.: Spoofing countermeasures to protect automatic speaker verification from voice conversion. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 3068–3072. IEEE (2013)
3. Barandela, R., Sánchez, J.S., Garcıa, V., Rangel, E.: Strategies for learning in class imbalance problems. Pattern Recognition **36**(3), 849–851 (2003)
4. Batista, G.E., Prati, R.C., Monard, M.C.: A study of the behavior of several methods for balancing machine learning training data. ACM Sigkdd Explorations Newsletter **6**(1), 20–29 (2004)
5. Batista, G.E.A.P.A., Prati, R.C., Monard, M.C.: A study of the behavior of several methods for balancing machine learning training data. SIGKDD Explor. Newsl. **6**(1), 20–29 (2004)
6. Batista, G.E.A.P.A., Prati, R.C., Monard, M.C.: Balancing strategies and class overlapping. In: Advances in Intelligent Data Analysis VI, 6th International Symposium on Intelligent Data Analysis, IDA 2005, Madrid, Spain, September 8-10, 2005, Proceedings, *LNCS*, vol. 3646, pp. 24–35. Springer (2005)
7. Bhowan, U., Johnston, M., Zhang, M.: Developing new fitness functions in genetic programming for classification with unbalanced data. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on **42**(2), 406–421 (2012)
8. Bhowan, U., Johnston, M., Zhang, M., Yao, X.: Evolving diverse ensembles using genetic programming for classification with unbalanced data. Evolutionary Computation, IEEE Transactions on **17**(3), 368–386 (2013)
9. Campbell, W.M., Sturim, D.E., Reynolds, D.A.: Support vector machines using gmm supervectors for speaker verification. Signal Processing Letters, IEEE **13**(5), 308–311 (2006)
10. Charbuillet, C., Gas, B., Chetouani, M., Zarader, J.L.: Optimizing feature complementarity by evolution strategy: Application to automatic speaker verification. Speech Communication **51**(9), 724–731 (2009)
11. Chawla, N.V., Japkowicz, N., Kotcz, A.: Editorial: special issue on learning from imbalanced data sets. ACM Sigkdd Explorations Newsletter **6**(1), 1–6 (2004)
12. Chen, L., Lee, K.A., Ma, B., Guo, W., Li, H., Dai, L.R.: Exploration of local variability in text-independent speaker verification. Journal of Signal Processing Systems **82**(2), 217–228 (2016)
13. Curry, R., Lichodzijewski, P., Heywood, M.I.: Scaling genetic programming to large datasets using hierarchical dynamic subset selection. IEEE Transactions on Systems, Man, and Cybernetics: Part B - Cybernetics **37**(4), 1065–1073 (2007)
14. Dat, T.T., Kim, J.Y., Kim, H.G., Lee, K.R.: Robust speaker verification using low-rank recovery under total variability space. In: IT Convergence and Security (ICITCS), 2015 5th International Conference on, pp. 1–4. IEEE (2015)
15. Day, P., Nandi, A.K.: Robust text-independent speaker verification using genetic programming. Audio, Speech, and Language Processing, IEEE Transactions on **15**(1), 285–295 (2007)
16. Day, P., Nandi, A.K.: Evolution of superfeatures through genetic programming. Expert Systems **28**(2), 167–184 (2011)
17. Dehak, N., Dumouchel, P., Kenny, P.: Modeling prosodic features with joint factor analysis for speaker verification. Audio, Speech, and Language Processing, IEEE Transactions on **15**(7), 2095–2103 (2007)
18. Dehak, N., Kenny, P.J., Dehak, R., Dumouchel, P., Ouellet, P.: Front-end factor analysis for speaker verification. Audio, Speech, and Language Processing, IEEE Transactions on **19**(4), 788–798 (2011)
19. Dick, G., Rimoni, A.P., Whigham, P.A.: A re-examination of the use of genetic programming on the oral bioavailability problem. In: Proceedings of the 2015 on Genetic and Evolutionary Computation Conference, pp. 1015–1022. ACM (2015)
20. Doucette, J., Heywood, M.I.: GP classification under imbalanced data sets: Active subsampling and AUC approximation. In: Proceedings of EuroGP 2008, *LNCS*, vol. 4971, pp. 266–277. Springer (2008)
21. Drummond, C., Holte, R.C., et al.: C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling. In: Workshop on Learning from Imbalanced Datasets II, vol. 11. Citeseer (2003)

22. Eggermont, J., Eiben, A.E., van Hemert, J.I.: Adapting the fitness function in GP for data mining. In: GP, Second European Workshop, Göteborg, Sweden, May 26-27, 1999, Proceedings, *LNCS*, vol. 1598, pp. 193–202. Springer (1999)
23. Evans, N.W., Kinnunen, T., Yamagishi, J.: Spoofing and countermeasures for automatic speaker verification. In: INTERSPEECH, pp. 925–929 (2013)
24. Garofolo, J.S., Lamel, L.F., Fisher, W.M., Fiscus, J.G., Pallett, D.S.: Darpa timit acoustic-phonetic continous speech corpus cd-rom. nist speech disc 1-1.1. NASA STI/Recon Technical Report N **93**, 27,403 (1993)
25. Gathercole, C., Ross, P.: Dynamic training subset selection for supervised learning in genetic programming. In: Parallel Problem Solving from Nature III, *LNCS*, vol. 866, pp. 312–321. Springer-Verlag, Jerusalem (1994)
26. George, K.K., Kumar, C.S., Ramachandran, K., Panda, A.: Cosine distance features for robust speaker verification. In: Proc. Interspeech, pp. 234–238 (2015)
27. Goncalves, I., Silva, S., Melo, J.B., Carreiras, J.M.B.: Random sampling technique for overfitting control in genetic programming. In: Proceedings of EuroGP 2012, *LNCS*, vol. 7244, pp. 218–229. Springer Verlag, Malaga, Spain (2012)
28. Hasan, T., Hansen, J.H.: Maximum likelihood acoustic factor analysis models for robust speaker verification in noise. IEEE/ACM Transactions on Audio, Speech, and Language Processing **22**(2), 381–391 (2014)
29. Hermansky, H.: Perceptual linear predictive (plp) analysis of speech. The Journal of the Acoustical Society of America **87**, 1738 (1990)
30. Hermansky, H., Morgan, N., Bayya, A., Kohn, P.: Rasta-plp speech analysis technique. In: Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on, vol. 1, pp. 121–124. IEEE (1992)
31. Hodges, J., Lehmann, E.L., et al.: Rank methods for combination of independent experiments in analysis of variance. The Annals of Mathematical Statistics **33**(2), 482–497 (1962)
32. Holm, S.: A simple sequentially rejective multiple test procedure. Scandinavian journal of statistics pp. 65–70 (1979)
33. Holmes, J.H.: Differential negative reinforcement improves classifier system learning rate in two-class problems with unequal base rates. In: 3rd Annual Conf. on Genetic Programming, pp. 635–642. ICSC Academic Press (1998)
34. Huang, X., Acero, A., Hon, H.W., et al.: Spoken language processing, vol. 15. Prentice Hall PTR New Jersey (2001)
35. Japkowicz, N., Stephen, S.: The class imbalance problem: A systematic study. Intelligent data analysis **6**(5), 429–449 (2002)
36. Joder, C., Schuller, B.: Exploring nonnegative matrix factorization for audio classification: Application to speaker recognition. In: Speech Communication; 10. ITG Symposium; Proceedings of, pp. 1–4. VDE (2012)
37. Kenny, P., Boulianne, G., Ouellet, P., Dumouchel, P.: Factor analysis simplified. In: Proc. ICASSP, vol. 1, pp. 637–640. Citeseer (2005)
38. Kenny, P., Boulianne, G., Ouellet, P., Dumouchel, P.: Joint factor analysis versus eigenchannels in speaker recognition. Audio, Speech, and Language Processing, IEEE Transactions on **15**(4), 1435–1447 (2007)
39. Kenny, P., Stafylakis, T., Ouellet, P., Alam, M.J., Dumouchel, P.: Plda for speaker verification with utterances of arbitrary duration. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 7649–7653. IEEE (2013)
40. Kinnunen, T., Hautamäki, V., Fränti, P.: Fusion of spectral feature sets for accurate speaker identification. In: 9th Conference Speech and Computer (2004)
41. Kinnunen, T., Li, H.: An overview of text-independent speaker recognition: From features to supervectors. Speech communication **52**(1), 12–40 (2010)
42. Kinnunen, T., Saeidi, R., Sedlák, F., Lee, K.A., Sandberg, J., Hansson-Sandsten, M., Li, H.: Low-variance multitaper mfcc features: a case study in robust speaker verification. IEEE Transactions on Audio, Speech, and Language Processing **20**(7), 1990–2001 (2012)
43. Kinnunen, T., Wu, Z.Z., Lee, K.A., Sedlak, F., Chng, E.S., Li, H.: Vulnerability of speaker verification systems against voice conversion spoofing attacks: The case of telephone speech. In: 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4401–4404. IEEE (2012)

44. Kubat, M., Matwin, S.: Addressing the curse of imbalanced training sets: One-sided selection. In: D.H. Fisher (ed.) Proceedings of the Fourteenth International Conference on Machine Learning (ICML 1997), Nashville, Tennessee, USA, July 8-12, 1997, pp. 179–186. Morgan Kaufmann (1997)
45. Lartillot, O., Toiviainen, P.: A matlab toolbox for musical feature extraction from audio. In: International Conference on Digital Audio Effects, pp. 237–244 (2007)
46. Li, M., Kim, J., Lammert, A., Ghosh, P.K., Ramanarayanan, V., Narayanan, S.: Speaker verification based on the fusion of speech acoustics and inverted articulatory signals. Computer Speech & Language **36**, 196–211 (2016)
47. Liu, X.Y., Wu, J., Zhou, Z.H.: Exploratory undersampling for class-imbalance learning. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on **39**(2), 539–550 (2009)
48. Liares, L.R., Garca-Mateo, C., Alba-Castro, J.L.: On combining classifiers for speaker authentication. Pattern Recognition **36**(2), 347–359 (2003)
49. Logan, B., et al.: Mel frequency cepstral coefficient for music modelling. In: ISMIR (2000)
50. Loughran, R., Agapitos, A., Kattan, A., Brabazon, A., O'Neill, M.: Speaker verification on unbalanced data with genetic programming. In: Applications of Evolutionary Computation, pp. 737–753. Springer (2016)
51. Loughran, R., Walker, J., ONeill, M., McDermott, J.: Genetic programming for musical sound analysis. In: Evolutionary and Biologically Inspired Music, Sound, Art and Design, pp. 176–186. Springer (2012)
52. Loughran, R.B.: Musical instrument identification with feature selection using evolutionary methods. Ph.D. thesis, University of Limerick (2009)
53. Makhoul, J.: Linear prediction: A tutorial review. Proceedings of the IEEE **63**(4), 561–580 (1975)
54. Márquez-Vera, C., Cano, A., Romero, C., Ventura, S.: Predicting student failure at school using genetic programming and different data mining approaches with high dimensional and imbalanced data. Applied intelligence **38**(3), 315–330 (2013)
55. Meriem, F., Farid, H., Messaoud, B., Abderrahmene, A.: Robust speaker verification using a new front end based on multitaper and gammatone filters. In: Signal-Image Technology and Internet-Based Systems (SITIS), 2014 Tenth International Conference on, pp. 99–103. IEEE (2014)
56. O'Shaughnessy, D.: Speech communication: human and machine. Universities press (1987)
57. Reynolds, D.A., Quatieri, T.F., Dunn, R.B.: Speaker verification using adapted gaussian mixture models. Digital signal processing **10**(1), 19–41 (2000)
58. Saeidi, R., Lee, K.A., Kinnunen, T., Hasan, T., Fauve, B., Bousquet, P.M., Khoury, E., Sordo Martinez, P., Kua, J.M.K., You, C., et al.: I4u submission to nist sre 2012: A large-scale collaborative effort for noise-robust speaker verification (2013)
59. Sivaram, G.S., Thomas, S., Hermansky, H.: Mixture of auto-associative neural networks for speaker verification. In: INTERSPEECH, pp. 2381–2384 (2011)
60. Song, D., Heywood, M.I., Zincir-Heywood, A.N.: Training genetic programming on half a million patterns: an example from anomaly detection. Evolutionary Computation, IEEE Transactions on **9**(3), 225–239 (2005)
61. Variani, E., Lei, X., McDermott, E., Moreno, I.L., Gonzalez-Dominguez, J.: Deep neural networks for small footprint text-dependent speaker verification. In: 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4052–4056. IEEE (2014)
62. Winkler, S.M., Affenzeller, M., Wagner, S.: Advanced genetic programming based machine learning. J. Math. Model. Algorithms **6**(3), 455–480 (2007)
63. Wu, Z., Evans, N., Kinnunen, T., Yamagishi, J., Alegre, F., Li, H.: Spoofing and countermeasures for speaker verification: a survey. Speech Communication **66**, 130–153 (2015)
64. Wu, Z., Kinnunen, T., Evans, N., Yamagishi, J., Hanilçi, C., Sahidullah, M., Sizov, A.: Asvspoof 2015: the first automatic speaker verification spoofing and countermeasures challenge. Training **10**(15), 3750 (2015)
65. Xue, B., Zhang, M., Browne, W., Yao, X.: A survey on evolutionary computation approaches to feature selection (2015)



(a) (b)

(c) (d)

(e) (f)

(g) (h)

Fig. 4 Comparison of the Test Classification accuracy on each of the Machine Learning Classifiers using Original Selection and Accuracy Selection.