TWO-STAGE PHONE DURATION MODELLING WITH FEATURE CONSTRUCTION AND FEATURE VECTOR EXTENSION FOR THE NEEDS OF SPEECH SYNTHESIS

Alexandros Lazaridis, Todor Ganchev, Iosif Mporas, Evaggelos Dermatas
and Nikos Fakotakis

Artificial Intelligence Group, Wire Communications Laboratory,
Department of Electrical and Computer Engineering,
University of Patras, 26500 Rion-Patras, Greece
Tel. +30 2610 996496, Fax. +30 2610 997336
alaza@upatras.gr

ABSTRACT

We propose a two-stage phone duration modelling scheme, which can be applied for the improvement of prosody modelling in speech synthesis systems. This scheme builds on a number of independent feature constructors (FCs) employed in the first stage, and a phone duration model (PDM) which operates on an extended feature vector in the second stage. The feature vector, which acts as input to the first stage, consists of numerical and non-numerical linguistic features extracted from text. The extended feature vector is obtained by appending the phone duration predictions estimated by the FCs to the initial feature vector. Experiments on the American-English KED TIMIT and on the Modern Greek WCL-1 databases validated the advantage of the proposed two-stage scheme, improving prediction accuracy over the best individual predictor, and over a two-stage scheme which just fuses the first-stage outputs. Specifically, when compared to the best individual predictor, a relative reduction in the mean absolute error and the root mean square error of 3.9% and 3.9% on the KED TIMIT, and of 4.8% and 4.6% on the WCL-1 database, respectively, is observed. The improved accuracy of phone duration modelling contributes to better control of the prosody, and thus the quality of synthetic speech.

Keywords: Feature construction; phone duration modeling; statistical modeling; text-to-speech synthesis

1. INTRODUCTION

In a text-to-speech (TTS) system the accurate modelling and control of prosody leads to high quality synthetic speech. Prosody can be regarded as the implicit channel of information in the speech signal that conveys information about the expression of emphasis, attitude, assumptions and the affected state of the speaker that provides the listener clues to support the recovery of the verbal message [1]. Prosody is shaped by the relative level of the fundamental frequency, the intensity and, last but not least, by the duration of the pronounced phones [2,3]. The duration of the phones controls the rhythm and the tempo of speech [4]. Flattening the prosody in a speech waveform would result in a monotonous, neutral and toneless speech, without rhythm, sounding unnatural and unpleasant to the listener, and sometimes even scarcely intelligible [5]. Thus, the accurate modelling of the duration of the phones is essential in speech synthesis, contributing to the naturalness of synthetic speech and consequently to the quality of the speech [1,4,6-12].

Various studies concerning phone duration modelling [7-28] have been made over the last few decades. The existing phone duration modelling methods are divided into two major categories: rule-based [12-17] and data-driven methods [7-11,18-25]. The rule-based methods utilise manually produced rules which are extracted from experimental studies on large sets of utterances or are based on prior knowledge. The extraction of these rules requires linguistic expertise. One of the first and most well known attempts in the field of rule-based phone duration modelling is the one proposed in [12] for the English language. In this method, rules based on linguistic and phonetic information, such as positional and prosodic factors, were used in order to predict the duration of the phones. These rules were derived by analysing a phonetically balanced set of sentences. Initially, a set of intrinsic values was assigned to each phone which was modified each time according to the extracted rules. Similar models were developed in other languages and dialects such as French [13], Brazilian Portuguese [14], Swedish [15], German [16] and Greek [17]. The main disadvantage of the rule-based methods is the difficulty to represent and manually tune all the linguistic, phonetic and prosodic factors which influence the duration of the phones in speech. As a result, it is very difficult to collect all the appropriate (or even enough) rules without long-term commitment to this task [29]. Therefore, in order to be able to deduce the interaction among these factors and extract these rules, the rule-based duration models are restricted to controlled experiments, where only a limited number of contextual factors are involved [30].

The creation of large databases made the development of data-driven methods for the task of phone duration modelling possible [31]. Data-driven methods overcame the problem of manual rule extraction by employing machine learning techniques that automatically produce phonetic rules and construct duration models from large speech corpora. The classical approach, which can be summarised by the process shown in Fig. 1(*a*), relies on features extracted from a database which are then projected onto the phone duration space through a machine learning method. The main advantage of data-driven methods in comparison to rule-based methods is that this process significantly reduces the efforts (manual work) of linguists.

Figure 1

The present work was inspired by studies on processing, combining or transforming an initial feature set into a new set of features [32-43]. This procedure, referred to as feature construction, can either lead to the reduction of the complexity of the feature space [44] or to the enrichment of the initial feature space with additional features [45,46]. Feature construction uses one or more operators (Boolean expressions, logical rules, etc.) to combine two or more features of the initial feature set to create a new feature, or to transform features of the initial feature set, thus creating new ones.

In the present work, we propose a phone duration modelling approach attempting to construct more accurate models which could lead to an improvement in the quality of synthetic speech. The proposed scheme incorporates a feature vector extension (FVE) stage, implemented through a feature construction process, illustrated in Fig. 1(*b*). In contrast to the classical approach (Fig. 1(*a*)), which directly uses feature vectors extracted from data to predict the duration of the phones, the proposed two-stage approach incorporating a FVE stage attempts to enrich the initial feature space with newly constructed features, and consequently predict the duration of the phones more accurately. This scheme builds on a number of feature constructors (FCs) employed in the first stage, and a phone duration model (PDM) in the second stage which operates on the extended feature vector. The FCs operate on a common input, the initial feature vector, which is constituted of linguistic features extracted only from text. The extended feature vector is obtained by appending the phone duration predictions estimated by the independent FCs to the initial feature vector. The newly constructed features capture the dependency between the initial features and the actual phones' durations in a manner that depends on the modelling technique and the training strategy employed in each of the FCs. In order for the FVE process to be beneficial in terms of overall phone duration prediction accuracy, it is essential that the FCs are based on various machine learning techniques. These techniques must rely on different assumptions and mapping functions of the feature space to the target space. In order to identify the most beneficial algorithm for the PDM, we evaluated ten different linear and non-linear models, which are based on linear regression, decision trees, meta-learning algorithms, lazy-learning algorithms, support vector machines and neural networks. As far as we are aware, the two-stage phone duration modelling scheme investigated in the present work has not been studied previously in the phone duration modelling task.

The remainder of this article is organised as follows. Related work concerning phone duration modelling methods and features used in this task are overviewed in Section 2. In Section 3 we outline the proposed two-stage phone duration modelling scheme. In Section 4 we briefly describe the phone duration modelling algorithms which are used to implement the independent FCs. These algorithms are also used along with some additional algorithms for implementing the PDM employed in the second stage. In addition, in Section 4 we briefly outline the databases and the experimental protocol followed in the numerical evaluation. The evaluation results are presented and discussed in Section 5. Section 6 concludes this work with a summary of the benefits of the proposed two-stage phone duration modelling scheme.

2. RELATED WORK

2.1 Phone duration modelling methods

In this section, we briefly report on some of the most frequently and successfully used methods in phone duration modelling (cf. Table 1). The linear regression (LR) [4,7,18,20,25] models are based on the assumption that there is linear independency among the features which affect phone duration. Specifically, the features are weighted in a linear combination creating a prediction function. On the other hand, decision tree-based models, and in particular classification and regression tree (CART) models [4,7-9,11,18,21,22,25], can represent the dependencies among the features but cannot put constraints on linear independency for reliable predictions [7]. Model trees, which is another tree-based technique, overcome the drawbacks of each of these two methods, and incorporates both linear regression and regression trees [4,7,25].

Moreover, Bayesian networks (BN) models have also been applied to the phone duration modelling task, incorporating a straightforward representation of the problem domain information. Despite their demanding training, it was shown that they make accurate predictions even when unknown values occur in some features [8]. In addition, artificial neural networks, such as the feed-forward neural networks, have been used in phone duration modelling [19]. Lazy-learning algorithms [18] have also been used in this task. In these algorithms the training data are stored during the training phase and a distance function is utilising during the prediction phase in order to determine which member of the training set is closer to the test instance and predict the phone duration. Furthermore, the sums-of-products (SOP) method has been used in phone duration modelling. In this approach the prediction of phones' durations is achieved based on a sum of factors and their product terms that affect the duration [8,22,23].

In a recent study [4], the gradient tree boosting (GTB) [47,48] method was applied to this task as an alternative to the conventional method using regression trees. GTB is a meta-learning algorithm which is based on the construction of multiple regression trees and consequently takes advantage of them. Moreover in [25], support vector regression (SVR) was applied to the task of phone duration modelling. Finally in the same work [25] a fusion scheme of different phone duration models, operating in parallel, was also proposed. Specifically, the predictions from a group of independent phone duration models were fed into a machine learning algorithm, which reconciled and fused the outputs of these models, improving the accuracy of the system. The data-driven methods provide the mechanism for overcoming the time consuming labour which is needed for the manual extraction of rules in rule-based phone duration modelling methods.

Table 1

2.2 Features used in the task of phone duration modelling

The features used in phone duration modelling are extracted from text and belong to various levels of representation of speech, such as the phonetic, phonological, morphological and syntactic levels. Some of the most frequently used features in this task are the phone, the number of phones in the syllable, the stress of the syllable, the position of the phone in the syllable, the position of the syllable in the word or in the phrase and the part-of-speech of the word. In addition, in some studies [7,10,18] apart from the stress feature, more prosodic features were used in the phone duration modelling task, such as the accent of the syllable, the type of the syllable, the distance to the next or previous accent, the break after the syllable and the distance to the next break or pause. Furthermore, in [4,8-10,18], features concerning the phonetic characteristics of the phones have been incorporated in this task such as the vowel height, the vowel frontness, the lip rounding, the manner of production and the place of articulation, or the number of phones before or after the vowel of the syllable. In other studies [8,9,18,19], information concerning the neighbouring instances, such as the next or previous phones, the type of the next or previous syllable, the stress or accent of the next or previous syllable, was taken into consideration. In the present work we take advantage of these previous studies and use a large feature vector that includes all of the above mentioned features.

3. THE TWO-STAGE PHONE DURATION MODELLING SCHEME

The two-stage phone duration modelling scheme is based on the use of multiple independent FCs in the first stage and a single PDM in the second stage, which operates on an extended feature vector. The phone duration predictions estimated by the independent FCs in the fisrt stage, are appended to the initial feature vector, creating the extended feature vector which is used by the PDM in the second stage predicting the duration of the phones. This two-stage scheme is based on the following steps:

(i) the independent FCs demonstrate different performances in the phone duration modelling task, i.e. they err in a different manner. In order for this statement to be valid, we specify two criteria for selecting the PDMs that will serve as FCs: (a) the candidate models have to demonstrate state-of-the-art performance and/or to have the advantage of a specific category of units, and (b) the input feature vector, the machine learning technique or the training strategy have to be different from the other models that have already been selected for the first stage.

(ii) the proposed two-stage scheme is expected to offer advantageous phone duration prediction accuracy, when compared to the best individual model, due to the benefits of fusion. These benefits are based on the known advantage that fusion of scores of multiple predictors offers [49-53].

(iii) the proposed two-stage scheme with FVE would improve the phone duration prediction accuracy when compared to a baseline two-stage scheme which just fuses the outputs of the first stage predictors, i.e. without implementing FVE, as proposed in [25]. This advantage should come from extending the initial feature vector with the newly constructed features. These features convey information about the mapping of the initial feature space to the target "phone duration" space, since they offer independent projections between these two spaces. These projections, although they might be considered as noisy, error-prone, imperfect and correlated with the initial feature vector, are expected to facilitate the PDM in the second stage. This is due to their different perspective and to the independent mapping between the input and output, which assists in correcting anomalies.

In Fig. 2, we present the block diagram of the proposed two-stage phone duration modelling scheme. Following the criteria formulated in (i), we assume the availability of numerous independent PDMs operating as FCs. As mentioned previously, the FCs should be implemented with different machine learning techniques or trained with different strategies and could operate on different subsets of features, which represent the data in a complementary way or on a common input feature vector. As presented in the figure, in the following we consider the case where all the FCs operate on a common input. This feature vector is composed of linguistic features extracted only from text (cf. Section 4.3) since text is the input in TTS systems. In the following subsections we describe the FCs and the PDMs in detail.

Figure 2

3.1 *Feature constructors* (*FCs*)

The training of the proposed two-stage phone duration modelling scheme depends on two non-overlapping datasets: the training and the development sets. Initially, the independent FCs are trained using the training dataset, and consequently the trained FCs are used to process the development dataset in order to produce new features. These newly constructed features, which are in fact phone duration predictions, are appended to the initial feature vector, which contains linguistic features of several speech representation levels, e.g. phonetic, phonological and morphosyntactic. The composite feature vector obtained after this merging, referred to as the extended feature vector, is employed for the training of the PDM in the second stage of the scheme.

The training of the FCs can be formalised as follows: Let us define a set of N independent FCs, which operate on a common input. Furthermore, let us define a M-dimensional feature vector, $\mathbf{x}_j^p$, which consists of numeric and non-numeric features. Here, the subscript and superscript indexes of $\mathbf{x}_j^p$ stand for the instance j of training data for phone p. The feature vectors, extracted from the training dataset, are used together to train each FC. The trained FCs are then used to process the data of the development dataset and the outcome of this processing are phone duration predictions, $y_j^{n,p} = B_p^n\left(\mathbf{x}_j^p, b_n\left(p\right)\right)$, which serve as the newly constructed features. Here, each of the N outputs, $y_j^{n,p}$ (n=1…N, where n is the index of each FC) is a real number corresponding to predicted phone duration value, while B_p^n and $b_n\left(p\right)$ stand for the trained FC and the phone-dependent parameters of the nth FC, for phone p, respectively. The outputs of the FCs are concatenated to form the N-dimensional feature vector $\mathbf{y}_j^p = \left[y_j^{1,p},...,y_j^{n,p},...,y_j^{N,p}\right]$, which is next appended to the initial feature vector, $\mathbf{x}_j^p$, to obtain the extended feature vector acting as input to the second stage.

3.2 *Phone duration model* (*PDM*) *in the second stage*

The PDM in the second stage is trained with the extended feature vectors obtained after the processing of the development dataset by the FCs in the fist stage. In detail, the initial feature vector, $\mathbf{x}_j^p$, and the outputs of the N FCs, $\mathbf{y}_j^p$, are concatenated to form the extended feature vector $\mathbf{z}_j^p = \left[\mathbf{x}_j^p, \mathbf{y}_j^p\right]^T$ with dimensionality $L = M + N$. The extended feature vector $\mathbf{z}_j^p$ together with the ground truth labels from the database are used for training the PDM, denoted as F_{DM}. Once the F_{DM} is trained, the two-stage phone duration modelling scheme is ready for operation.

During run-time operation of the two-stage scheme (cf. Fig. 2), the input data are processed as follows: An M-dimensional input feature vector $\mathbf{x}_j^p$, for the jth instance, appears as input to all the FCs. The outputs of the FCs, i.e. the phone duration predictions, $y_j^{n,p}$, are the newly constructed features. Consequently, they are appended to the initial feature vector $\mathbf{x}_j^p$ used in the first stage, and the L-dimensional extended feature vector obtained from this, is fed as input into the PDM in the second stage. The output of the model F_{DM} is the final phone duration prediction, $o_j^p = F_{DM}^p\left(\mathbf{z}_j^p, f_n\left(p\right)\right)$, where F_{DM}^{p} is the constructed duration model for phone *p*, and $f_n(p)$ are the phone-dependent parameters of F_{DM} .

4. EXPERIMENTAL SETUP

In order to investigate the practical usefulness of the proposed two-stage scheme, we trained and evaluated a number of independent models, which were employed to operate as FCs in the first stage. Furthermore we investigated various implementations of the PDM in the second stage. The machine learning algorithms used in the implementation of the FCs and PDM are described in the following two subsections.

4.1 Feature construction algorithms

We considered eight independent phone duration modelling methods for use as FCs, which are well known and have been successfully used over the years in different modelling tasks. These are:

(i) linear regression (*LR*) [54] using Akaike's Information Criterion (AIC) [55] in a backward stepwise selection (BSS) [56] procedure to eliminate unnecessary variables of the training data. The basic idea of the linear regression algorithm is to express the prediction values (i.e. the phones' durations) as a linear combination of the features with weights which are determined during the training phase of the algorithm.

(ii) two decision trees: a model tree (*m5p*) and a regression tree (*m5pR*) [57,58]. Decision trees are predictive models that create a mapping procedure between observations about an item and the conclusions about its target value. In these tree structures, leaves represent target values and branches represent conjunctions of features that lead to these target values. The main difference between these two algorithms is that a model tree utilises a linear regression function on each leaf, and alternatively a regression tree utilises a constant value on each leaf node [57,58].

(iii) two additive regression algorithms (*Add. Regr*.) [48] and two bagging (*Bagg*.) algorithms [59] were used, by utilising two different regression trees (*m5pR* and *REPTrees*) [57,58,60] as base classifiers in each case. The last four algorithms are meta-learning algorithms [61] using regression trees as base classifiers.

During the training procedure, in each iteration, the additive regression algorithm builds a regression tree using the residuals of the previous tree as training data. Since each tree added to the model fits the training data more closely, this procedure is prone to overfitting. In order to avoid overfitting of the model, instead of subtracting a model's entire prediction to generate the target values (residuals) for the next model, a constant factor (shrinkage) was used. This factor, taking values between 0 and 1, was used to shrink the predictions by multiplying them with it before subtracting. In this way the fitting of the model to the residuals is reduced, decreasing the chance of overfitting [54]. The regression trees are combined together creating the final prediction function. In these two cases of additive regression meta-classification the shrinkage parameter, v, indicating the learning rate, was set equal to 0.5. Furthermore the number of the regression trees, *rt-num*, which were trained iteratively using the residuals of the tree of the previous iteration was set equal to 10. The values of both of these parameters were selected after a number of grid search experiments (v={0.1, 0.3, 0.5, 0.7, 0.9}, *rt-num*={5, 10, 15, 20}) on a randomly selected subset of the training dataset, representing 20% of the size of the full training dataset.

In the bagging algorithm, the dataset is split into multi subsets utilising one regression tree for each one of them. The final prediction value is the average of the values predicted from each regression tree. Also in this case, the number of the regression trees, *rt-num*, which were trained independently using each subset of the split dataset was set equal to 10 after grid search experiments (*rt-num*={5, 10, 15, 20}) on the randomly selected subset of the training dataset mentioned above.

(iv) Finally, the support vector regression (SVR) model [62] was used, which implements the sequential minimal optimisation (SMO) algorithm for training a support vector classifier (*SMOreg*) [63]. To this end, various kernel functions have been used in SVR, such as polynomial, radial basis function (RBF), Gaussian functions, etc. In our experiments the RBF kernel was used as the mapping function [64]. The basic idea governing the SVR is the production of a model that can be expressed through support vectors. A linear regression function is used to approximate the training instances by minimising the prediction error. A user-specified parameter ε defines a tube around the regression function. In this tube the errors are ignored. The parameter ε controls how closely the function will fit the training data. The parameter C is the penalty for exceeding the allowed deviation defined by ε. The larger the C, the more closely the linear regression function can fit the data [54]. The ε and C parameters, where $\varepsilon \geq 0$ and $C > 0$, were set equal to 10^{-3} and 10^{-1} respectively, after a grid search (ε={10^{-1}, 10^{-2}, …, 10^{-5}}, C={0.05, 0.1, 0.3, 0.5, 0.7, 1.0, 10, 100}) on the randomly selected subset of the training dataset mentioned above.

Our motivation to select these machine learning algorithms was based on previous research [4,7,20,21,24], where these algorithms were reported to be successful for phone duration modelling. Along with this task, many of these algorithms have also been used in syllable duration modelling tasks, supporting different languages and databases. Once these eight models are built with different machine learning techniques and have different training strategies, they will conform to the criteria formulated in Section 3.

4.2 Phone duration algorithms in the second stage

In order to select the most advantageous algorithm for the PDM in the second stage, we experimented on ten different machine learning algorithms. These included the eight algorithms outlined in Section 4.1, along with (i) the radial basis function neural network (*RBFNN*) with Gaussian kernel [65], and (ii) the instance-based algorithm (*IBK*) [66] which is a *k*-nearest neighbours classifier.

The *RBFNN* implements a Gaussian radial basis function network, deriving the centres and widths of hidden units using *k*-means [67] and combining the outputs obtained from the hidden layer using logistic regression if the class is nominal, and linear regression if it is numeric. The activations of the basis functions are normalised to sum to unity before they are fed into the linear models. The number of clusters and the minimum standard deviation for the clusters are the free parameters of the algorithm [54]. The number of clusters, *num-cl*, to be generated by the *k*-means algorithm and the minimum standard deviation, *cl-std*, for the clusters were set equal to 135 and 10^{-2}, respectively. These parameters were determined by a grid search (*num-cl*={5,10, …, 200}, *cl-std*={0.001, 0.01, 0.1, 0.5}) on the randomly selected subset of the training dataset representing 20% of the size of the full training dataset.

The *IBK* is an instance-based algorithm [66], which belongs to the lazy-learning algorithms. In the training phase it stores the training instances verbatim, and in the prediction phase it searches for the instance that most closely resembles the target instance in order to predict the target value. This is calculated through the use of a distance function. In the present work, we used the linear nearest neighbours search algorithm with Euclidean distance as the distance function. Leave-one-out cross-validation was used to select the best *k* value, with an upper limit of 35 nearest neighbours. The predictions from nearest neighbours were weighted according to the inverse distance.

4.3 Databases and feature set

The evaluation for the phone duration modelling task was carried out on two databases: the American English speech database, CSTR US KED TIMIT [68], and the Modern Greek speech prosodic database, WCL-1 [69]. The KED TIMIT database consists of 453 phonetically balanced sentences (3400 words approximately) uttered by a Native American male speaker. The WCL-1 prosodic database consists of 5500 words distributed in 500 paragraphs, each one of which may be a single word, a short or long sentence, or a sequence of sentences uttered by a female professional radio actress. The corpus includes 390 declarative sentences, 44 exclamation sentences, 36 decision questions and 24 “wh” questions.

For the experiments with the KED TIMIT database we made use of the phone set provided with the database [68] which consists of 44 phones. For the experiments using the WCL-1 database we made use of the phone set provided with the database [69] consisting of 34 phones. In all the experiments with both databases, the manually labelled durations of the phones were used as the ground truth reference durations.

In the present work, we consider all the features which have been reported [4,7-10,18-29] to have been used successfully in the task of phone duration modelling. Since the input of a TTS system is text, the initial feature vector is composed of linguistic features extracted only from text. In particular, from each utterance of the database, for each instance of the utterance which corresponds to a phone, we computed 33 features. The temporal neighbours of some of these features, defined on the level of the respective feature, i.e. phone-level, syllable-level and word-level, were also used. The features involved in the initial feature vector are as follows:

(i) eight phonetic features: the phone class (consonants/non-consonants), the phone types (vowels, diphthongs, schwa, consonants), the vowel height (high, middle or low), the vowel frontness (front, central or back), the lip rounding (rounded/unrounded), the manner of production (plosive, fricative, affricate, approximant, lateral, nasal), the place of articulation (labial, labiodental, dental, alveolar, palatal, velar, glottal), and the consonant voicing. Along with the aforementioned features, which concern each current instance, the corresponding information concerning the two previous and the two next instances (temporal context information) was also used.

(ii) three phone-level features: the phone name with the temporal context information of the neighbouring instances (previous, next), the position of the phone in the syllable and the onset-coda type (onset: if the specific phone is before the vowel in the syllable, coda: if the specific phone is the vowel or if it is after the vowel in the syllable).

(iii) thirteen syllable-level features: the position type of the syllable (single, initial, middle or final) with the information of the neighbouring instances (previous, next) and the number of all the syllables. Furthermore, the number of the accented syllables and the number of the stressed syllables since the last and to the next phrase break (i.e. the break index tier of ToBI [70] with values, 0, 1, 2, 3, 4,) were included. Moreover, the syllables' onset-coda size (the number of phones before and after the vowel of the syllable) with the information of previous and next instances, the onset-coda type (if the consonant before and after the vowel in the syllable is voiced or unvoiced), along with the temporal context information of previous and next instances, were used as syllable-level features. Finally, the position of the syllable in the word and the onset-coda consonant type (the manner of production of the consonant before and after the vowel in the syllable) were included.

(iv) two word-level features: the part-of-speech (noun, verb, adjective, etc.) and the number of syllables of the word.

(v) one phrase-level feature: the syllable break (the phrase break after the syllable) with the temporal context information of the neighbouring (two previous, two next) instances. The syllable break feature is implemented based on the break index tier of ToBI (0, 1, 2, 3, 4).

(vi) six accentual features: the ToBI [70] accents and boundary tones with the temporal context information of the neighbouring (previous, next) instances and the last-next accent (the number of the syllables since the last and to the next accented syllable) were used. Additionally, we included the stressed-unstressed syllable feature (whether or not the syllable is stressed) and the accented-unaccented syllable feature (whether or not the syllable is accented) with the information of the neighbouring (two previous, two next) instances.

The overall size of the initial feature vector is 93, including the aforementioned features and their temporal context information as reported above (one or two previous and next instances on the level of the respective feature, phone-level, syllable-level and word-level).

4.4 Experimental protocol

For the purpose of comparison, we evaluated the phone duration prediction accuracy of the proposed two-stage scheme with FVE in contrast with (i) the accuracy of the best independent FC method, and (ii) the fusion scheme proposed in [25], which in this work is referred to as fusion of the constructed features (FCF). The FCF scheme is equivalent to a direct fusion of the predictions of the FCs, since only the outputs of the FCs compose the feature vectors used as input for the PDM in the second stage. This scheme will be considered as the baseline to which the performance of the proposed FVE scheme is compared.

In all experiments we followed an experimental protocol based on 10-fold cross-validation. Specifically, in each fold the training data were split in two non-overlapping portions: the training dataset and the development dataset. The training dataset, amounting to approximately 60% of the full dataset, was utilised in the training of the first-stage models, the FCs, and the development dataset, amounting to approximately 30% of the full dataset, was used in the training of the PDM in the second stage. Furthermore, the test dataset, amounting to approximately 10% of the full dataset, was used for evaluating the performance of the eight individual FCs, as well as the overall performance of the two-stage scheme.

4.5 Performance metrics

Phone duration modelling, which mainly relies on regression algorithms, suffers from specific types of errors. The most commonly occurring type of error is the bias (systematic) error [71]. This error is a constant shift of the predicted phones' durations from the real ones and can be estimated as the difference between the real and predicted mean durations. Other prediction errors that may occur in the phone duration modelling task are small miss-predictions and gross errors (outliers) [71]. Small miss-predictions in phone duration, i.e. less than 20 milliseconds, do not significantly affect the quality of the synthetic speech signal. In contrast to these, the other errors degrade the quality of synthetic speech [72].

The experimental results are reported in terms of the two most commonly used figures of merit, namely the mean absolute error (MAE) and the root mean squared error (RMSE), between the predicted duration and the actual (reference) duration of each phone [4-6,8,9,11,22]. Due to the squaring of values in the RMSE the large errors (outliers) are weighted heavily, which makes this figure of metric more sensitive to outliers than the MAE [54]. This sensitivity of the RMSE makes it a more illustrative measurement concerning the gross errors, when compared to the MAE.

5. EXPERIMENTAL RESULTS

Three different ways of grouping the instances (i.e. the phones) of the database were considered, on the basis of (i) vowels/consonants categorisation of the phones, (ii) phonetic categories of the consonants and (iii) individual phones. These divisions of the data offer different degrees of detail and allow us to verify the reasoning behind our proposed scheme in various conditions. In brief, the main concept (cf. Section 3) was that (i) independent phone duration predictors implemented through different machine learning algorithms perform differently in different conditions and (ii) independent phone duration predictors can serve as FCs, which contribute to an improvement in overall phone duration prediction accuracy when they are involved in the proposed FVE scheme.

In the following subsections we present the results of the experimental evaluation of the accuracy of the eight individual FCs described in Section 4.1. Afterwards, these models are used in the first stage of the proposed two-stage scheme constructing the new features which are subsequently used for the FVE. Following which, we present results from the evaluation of the applicability of ten different phone duration modelling algorithms (see Section 4.2), employed in the second stage of the proposed two-stage phone duration prediction scheme.

5.1 Classical approach for phone duration modelling

In order to evaluate the accuracy of various phone duration prediction methods outlined in Section 4.1, which implement the classical approach for phone duration modelling (cf. Fig. 1(*a*)), we examined the performance of the eight FCs on both databases using the initial feature set described in Section 4.3. The RMSE, the MAE and the standard deviation of the absolute error (STD of AE) for all the FCs specified in Section 4.1 are shown in Table 2, where Table 2(*a*) presents the results obtained on the KED TIMIT database and Table 2(*b*) on the WCL-1 database. The results of the best performing model, among the eight FCs, are in bold. As it is shown in the table, the support vector machine (SVM) models, implemented with the SMO regression (*SMOreg*) model, outperform all the other models on both databases. Specifically, on the KED TIMIT database the *SMOreg* model outperforms the second-best model, which is the meta-classifier additive regression using m5pR (*Add. Regr. m5pR*) model, by 5.5% and 3.7% in terms of MAE and RMSE, respectively. On the WCL-1 database the *SMOreg* model outperforms the second-best model, *LR*, by 6.8% and 3.7% in terms of MAE and RMSE, respectively. This advantage of the *SMOreg* model is explained by the ability of SVMs to cope better with high-dimensional feature space [73,74], when compared to the other algorithms under consideration.

Table 2

In addition, in Table 3 we present the performance of the same eight FCs on the KED TIMIT (Table 3(*a*)) and the WCL-1 databases (Table 3(*b*)) for the case of vowel/consonant division and per phonetic category of the consonants. Again, the *SMOreg* model demonstrated the lowest RMSE on both databases, except for one case: the Affricates on KED TIMIT, where the lowest RMSE is observed for the meta-classifier additive regression using the *REPTrees*, where the *SMOreg* model has the second-best performance.

Table 3

In Table 4 the results of the FCs are analysed to the level of individual phones. Specifically, Table 4(*a*) reports the RMSE for the 44 phone set on the KED TIMIT database and Table 4(*b*) for the 34 phone set on the WCL-1 database. Again the results for the best performing algorithm are in bold. As shown in the tables, despite the fact that the *SMOreg* models demonstrate the highest overall performance on both databases (refer to Table 2), in one phonetic category (Affricates in Table 3(*a*)) and in some cases of individual phones (Table 4) other models offer a higher phone duration prediction accuracy. For instance, on the KED TIMIT database, for the phone *ch*, the *LR* model expressed the best performance, while for the phone *ay*, the *m5p* model offers the best performance (refer to Table 4 (*a*)). These two specific cases (and other similar ones reported in the tables) support the motivation behind our two-stage scheme, that different algorithms perform better in different phonetic categories, and therefore can improve the overall accuracy of phone duration modelling.

Table 4

5.2 Two-stage phone duration modelling with feature construction and feature vector extension

In this section, we investigate the accuracy of the proposed two-stage phone duration modelling scheme with FVE. This scheme is compared to the baseline scheme, i.e. FCF, where the initial feature vector is not propagated to the second stage and only the outputs of the FCs, which are the newly constructed features, are used as input to the PDM in the second stage. In the following we consider ten different implementations of the PDM in the second stage, and evaluate their performance both with and without FVE.

In Table 5, we present the results for the ten algorithms outlined in Section 4.2, which are the *LR*, the *m5p* model tree, the *m5pR* regression tree, the two additive regression algorithms based on *m5pR* and *REPTrees* (*Add. Regr. m5pR* and *Add. Regr. REPTrees*), the two bagging algorithms based on *m5pR* and *REPTrees* (*Bagg. m5pR* and *Bagg. REPTrees*), the instance-based learning (*IBK*), the support vector regression (SVR) with sequential minimal optimisation (SMO) training referred here as *SMOreg*, and the radial basis function neural network (*RBFNN*). In all cases the best results are shown in bold. Again for reasons of comparison, in Table 5 we also present the experimental results for the best individual FC (*SMOreg*).

Table 5

As identified in Table 5, the baseline FCF scheme, implemented with the *SMOreg* model, outperformed the best individual FC, the *SMOreg*, by 1.9% and 2% in terms of MAE and RMSE on the KED TIMIT database (Table 5(*a*)), and respectively by 2.6% and 1.8% on the WCL-1 database (Table 5(*b*)). The proposed FVE scheme outperformed the best FC (*SMOreg*) on both the KED TIMIT and WCL-1 databases only when the PDM in the second-stage is implemented with *SMOreg*. In terms of

MAE and RMSE, the benefit of the FVE is shown by the accuracy improvement of 3.9% and 3.9% on the KED TIMIT database and of 4.8% and 4.6% on the WCL-1 database, respectively.

Moreover, it should be mentioned that the proposed FVE scheme, apart from reducing the overall error, also reduces the deviation of the outliers. In the case when the error distribution is Gaussian, the reduction in the standard deviation of the absolute error is correlated to the reduction of the outliers with respect to the model. As the results in Table 5 show, the proposed FVE scheme reduced the STD of AE in comparison to the best FC, the *SMOreg*, by approximately 3.9% on the KED TIMIT database and by approximately 2.5% on the WCL-1 database, respectively.

As already stated, the FVE scheme outperforms the best individual FC, *SMOreg*, only when the PDM in the second-stage is implemented with *SMOreg*. This observation can be reasoned with the high dimensionality of the input feature vector (93 initial features + 8 newly constructed = 103 dimensions). This is because the other machine learning techniques, when employed as PDM in the second stage, do not build robust models from the available training data due to the curse of dimensionality. Since the support vector machines do not suffer from this problem they performed better than any of the other techniques evaluated here.

Finally, in order to investigate whether the differences in the accuracies between the best individual FC and the proposed FVE scheme, and between the baseline (FCF) and the proposed FVE scheme, are statistically significant, we performed the Wilcoxon test [75]. On both databases, the Wilcoxon test showed that these differences are statistically significant, with a significance level of *p*-value < 0.05. Thus, the proposed FVE scheme can be regarded as advantageous when compared to both the best individual FC and to the baseline FCF scheme.

5.3 Additional experiments with feature ranking

For further investigation of the effectiveness of the proposed FVE scheme, a subset selection on the extended feature vector was performed. For that purpose, we firstly performed feature ranking with the Recursive Elimination of Features (*RELIEF*) algorithm [76], and subsequently selected the top-20, top-50, top-80 ranked features. This resulted in three subsets of the extended feature vector, which in the following we refer to as Sets 20, 50 and 80, respectively. The entire extended feature vector is referred to as the *full feature set*.

The feature selection was performed on a randomly selected subset, corresponding to 40% of the development dataset. The eight constructed features obtained from the FCs (evaluated in Section 5.1), were ranked in the top-13 features on KED TIMIT, and in the top-18 on the WCL-1 database. Thus, the feature ranking results confirm the importance of the newly constructed features and gives an explanation for their contribution to the reduction of the error rates, when they are used as an extension of the initial feature vector.

Table 6

In Table 6, we present the results for the proposed FVE scheme, when the second stage is fed with the feature subsets 20, 50 and 80. Here Table 6(*a*) presents the results on the KED TIMIT database and Table 6(*b*) on the WCL-1 database. For reasons of comparison, in the last three columns of Table 6, we duplicate the experimental results for the full feature set. As can be seen in the table, in the case of Set 20, where only the top-20 features of the full feature set are fed to the second stage, the proposed FVE scheme with *SMOreg* in the second stage outperforms the best FC, the *SMOreg* model, by 2.7% and 2.7% in terms of MAE and RMSE respectively on the KED TIMIT database, and by 3% and 2.1% respectively on the WCL-1 database. However, the performance on Set 20 is worse than that obtained on the full feature set. The same is valid for Set 50 and Set 80. In the case of Set 50, where the top-50 features are fed to the second stage, once again only the proposed FVE scheme outperforms the best FC by 3.4% and 3.4% in terms of MAE and RMSE respectively on the KED TIMIT database, and by 3.6% and 3.3% respectively on the WCL-1 database. Likewise, for the Set 80, the proposed FVE scheme outperforms the best FC by 3.9% and 3.7% in terms of MAE and RMSE respectively on KED TIMIT, and by 4.4% and 4.1% respectively on the WCL-1 database.

Further analysing the results shown in Table 6, it should be noted that the models based on the *IBK* and the *RBFNN* techniques showed a noticeable drop in their accuracy as the number of the features in the feature vector was increased (from Set 20 to the full feature set), increasing the MAE by 10.3% and 7.5% on the KED TIMIT and WCL-1 databases, respectively, for *IBK* model, and by 28.9% and 18.7%, respectively, for the *RBFNN* model. This decline in the phone duration prediction accuracy shows that the *IBK* [77] and *RBFNN* [78] models suffer from the *curse of dimensionality* [79], and thus they do not handle well the increased dimensionality of the feature vector, given the predefined amount of the training data.

Furthermore, it is shown that for all the other machine learning techniques evaluated here, the increase in the feature vector dimensionality (from Set 20 to the full feature set) showed no significant improvement in the accuracy of the PDMs and in no case did any model outperform the best individual FC, *SMOreg*.

In conclusion, we can summarise that the experimental results confirmed the advantages of the proposed two-stage phone duration modelling scheme, which incorporates a number of FCs in the first stage and a SVM-based PDM in the second stage that operate on an extended feature vector. The proposed scheme contributes to a significant gain in accuracy, when compared (i) to the best individual FC, and (ii) to the case of simple fusion of the outputs of the FCs, i.e. without extension of the features. Finally, ranking the relevance of the individual features in the extended feature vector demonstrated the high relative importance of the newly constructed features, which explains the observed accuracy gain.

6. CONCLUSION

We studied a two-stage phone duration modelling scheme, which relies on a number of independent feature constructors employed in the first stage and a phone duration model in the second stage, operating on an extended feature vector. This scheme takes advantage of the fact that different prediction algorithms operating on a common input can construct complementary features, which when appended to the initial feature vector contribute to the improvement of the overall phone duration prediction accuracy. The support vector regression was found to be the most appropriate implementation for the second-stage phone duration model in the proposed scheme. Specifically, the SMO regression model outperformed the best feature constructor by 3.9% and 3.9% in terms of mean absolute error and root mean square error, respectively, on the KED TIMIT database and by 4.8% and 4.6%, respectively, on the WCL-1 database. The extended feature vector, consisting of the initial feature vector and the newly constructed features, was found to be advantageous over the three smaller subsets. The proposed two-stage scheme improved the accuracy of phone duration modelling, contributing to better control of the prosody.

7. REFERENCES

[1] X. Huang, A. Acero, H.W. Hon, Spoken Language Processing: a guide to theory, algorithm, and system development, Prentice Hall, 2001.

[2] T. Dutoit, An Introduction to Text-To-Speech Synthesis, Kluwer Academic Publishers, Dordrecht, 1997.

[3] S. Furui, Digital Speech Processing, Synthesis, and Recognition, second ed., Marcel Dekker, 2000.

[4] J. Yamagishi, H. Kawai, T. Kobayashi, Phone duration modeling using gradient tree boosting, Speech Communication 50:5 (2008) 405-415.

[5] S.H. Chen, W.H. Lai, Y.R. Wang, A new duration modeling approach for Mandarin speech, IEEE Trans. on Speech Audio Processing 11:4 (2003) 308-320.

[6] S.H. Chen, S.H. Hwang, Y.R. Wang, An RNN-based prosodic information synthesizer for Mandarin text-to-speech, IEEE Trans. on Speech Audio Processing 6:3 (1998) 226-239.

[7] N. Iwahashi, Y. Sagisaka, Statistical modeling of speech segment duration by constrained tree regression, IEICE Trans. Inform. Systems E83-D:7 (2000) 1550-1559.

[8] O. Goubanova, S. King, Bayesian networks for phone duration prediction, Speech Communication 50:4 (2008) 301-311.

[9] N.S. Krishna, H.A. Murthy, Duration modeling of Indian languages Hindi and Telugu, in Proc. of the 5th ISCA Speech Synthesis Workshop, Pittsburgh, USA, (2004) 197-202.

[10] B. Mobius, J. van Santen, Modeling segmental duration in German text-to-Speech synthesis, in Proc. of ICSLP-1996, Philadelphia, USA (1996) 2395-2398.

[11] S. Lee, Y.H. Oh, Tree-based modeling of prosodic phrasing and segmental duration for Korean tts systems., Speech Communication 28 (1999) 283-300.

[12] D.H. Klatt, Linguistic uses of segmental duration in English: Acoustic and perceptual evidence, Journal of the Acoustical Society of America 59 (1976) 1209-1221.

[13] K. Bartkova, C. Sorin, A model of segmental duration for speech synthesis in French, Speech Communication 6 (1987) 245-260.

[14] A.R.M. Simoes, Predicting sound segment duration in connected speech: An acoustical study of Brazilian Portugese, in Proc. of Workshop on Speech Synthesis (1990) 173-176.

[15] R. Carison, B. Granstrom, A search for durational rules in real speech database, Phonetica 43 (1986) 140-154.

[16] K.J. Kohler, Zeistrukturierung in der Sprachsynthese, ITG-Tagung Digitalc Sprachverarbeitung 6 (1988) 165-170.

[17] G. Epitropakis, D. Tambakas, N. Fakotakis, G. Kokkinakis, Duration modelling for the Greek language, in Proc. of EUROSPEECH (1993) 1995-1998.

[18] A. Lazaridis, P. Zervas, G. Kokkinakis, Segmental duration modeling for Greek speech synthesis, in Proc. of IEEE ICTAI (2007) 518-521.

[19] J.P. Teixeira, D. Freitas, Segmental durations predicted with a neural network, in Proc. of the European Conference on Speech Communication and Technology, Geneva, Switzerland, September, (2003) 169-172.

[20] K. Takeda, Y. Sagisaka, H. Kuwabara, On sentence-level factors governing segmental duration in Japanese, Journal of Acoustic Society of America 86:6 (1989) 2081-2087.

[21] M. Riley, Tree-based modelling for speech synthesis, in: G. Bailly, C. Benoit, T.R. Sawallis (Eds.), Talking Machines: Theories, Models and Designs, Elsevier, Amsterdam, Netherlands, 1992, pp. 265-273.

[22] H. Chung, Duration models and the perceptual evaluation of spoken Korean, in Proc. of Speech Prosody, France, (2002) 219-222.

[23] J.P.H. van Santen, Assignment of segment duration in text-to-speech synthesis, Computer Speech and Language 8:2 (1994) 95-128.

[24] S. Lee, Y. Oh, CART-based modelling of Korean segmental duration, in Proc. of Oriental COCOSDA (1999) 109-112.

[25] A. Lazaridis, I. Mporas, T. Ganchev, G. Kokkinakis, and N. Fakotakis, Improving Phone Duration Modeling using Support Vector Regression Fusion, Speech Communication, (In Press).

[26] T.H. Crystal, A.S. House, Segmental durations in connected-speech signals: Current results, Journal of the Acoustical Society of America 83:4 (1988) 1553-1573.

[27] M. Gregory, A. Bell, D. Jurafsky, W. Raymond, Frequency and predictability effects on the duration of content words in conversation, Journal of the Acoustical Society of America 110:5 (2001) 27–38.

[28] A. Bell, D. Jurafsky, E. Fosler-Lussier, C. Girand, M. Gregory, D. Gildea, Effects of disfluencies, predictability, and utterance position on word form variation in English conversation, Journal of the Acoustical Society of America 113:2 (2003) 1001-1024.

[29] D.H. Klatt, Review of text-to-speech conversion for English, Journal of the Acoustical Society of America 82:3 (1987) 737-793.

[30] K.S. Rao, B. Yegnanarayana, Modeling durations of syllables using neural networks, Computer Speech & Language 21:2 (2007) 282-295.

[31] J. Kominek, A.W. Black, CMU ARCTIC databases for speech synthesis, CMU-LTI-03-177, Language Technologies Institute, School of Computer Science, Carnegie Mellon University, 2003.

[32] L. Breiman, J. Friedman, R. Olshen, C. Stone, Classification and Regression Trees, Belmont: CAWadsworth International Group, 1984.

[33] C.J. Matheus, L. Rendell, Constructive induction in decision trees, in Proc. of 11th IJCAI (1989) 645-650.

[34] G. Pagallo, D. Haussler, Boolean feature discovery in empirical learning, in Proc. of 7th International Conference on Machine Learning, San Mateo, CA: Morgan Kaufmann (1990) 71-99.

[35] L. Watanabe, L.A. Rendell, Feature construction in structural decision trees, in Proc. of 8th International Conference on Machine Learning, San Mateo, CA: Morgan Kaufmann (1991) 218-222.

[36] P.A. Flach, N. Lavrac, The role of feature construction in inductive rule learning, in Proc. of the International Conference on Machine Learning 2000, Workshop on Attribute-value and Relational Learning: Crossing the boundaries (2000) 1-11.

[37] K.A. De Jong, W.M. Spears, D.F. Gordon, Using genetic algorithms for concept learning, Machine Learning 8 (1992) 5-32.

[38] D. Heath, S. Kasif, S. Salzberg, Learning oblique decision trees, in Proc. of 13th International Conference on Artificial Intelligence (1993) 1003-1007.

[39] J.R. Quinlan, C4.5. Programs for Machine Learning, Morgan Kaufman, 1993.

[40] H. Ragavan, L.A. Rendell, M. Shaw, A. Tessmer, Complex concept acquisition through directed search and feature caching, in Proc. of 13th International Conference on Artificial Intelligence (1993) 946-951.

[41] J. Wenk, R.S. Michalski, Hypothesis-driven constructive induction in AQ17-HCI: A method and experiments, Machine Learning 14 (1994) 139-168.

[42] Y.J. Hu, D. Kibler, Generation of attributes for learning algorithms, in Proc. of 13th International Conference on Machine Learning, San Mateo, CA: Morgan Kaufmann (1996) 806-811.

[43] Z. Zheng, Constructing nominal x-of-n attributes, in Proc. of 13th International Conference on Machine Learning, San Mateo, CA: Morgan Kaufmann (1996) 1064-1070.

[44] S. Piramuthu, R.T. Sikora, Iterative feature construction for improving inductive learning algorithms, Expert Systems with Applications 36:2 (2009) 3401-3406.

[45] D. Koller, M. Sahami, Toward optimal feature selection, in Proc. of 13th International Conference on Machine Learning (1996) 129-134.

[46] M. Dash, H. Liu, Feature selection for classification, International Journal of Intelligent Data Analysis 1 (1997) 131-156.

[47] J.H. Friedman, Greedy function approximation: a gradient boosting machine, Annals of Statistics 29:5 (2001) 1189-1232.

[48] J.H. Friedman, Stochastic gradient boosting, Computational Statistics & Data Analysis 38:4 (2002) 367-378.

[49] D.H. Wolpert, Stacked generalization, Neural Networks 5:2 (1992) 241-260.

[50] S. Hashem, B. Schmeiser, Improving model accuracy using optimal linear combinations of trained neural networks, IEEE Trans. on Neural Networks 6:3 (1995) 792-794.

[51] J. Kittler, M. Hatef, R.P.W. Duin, J. Matas, On combining classifiers, IEEE Trans. on Pattern Analysis and Machine Intelligence 20:3 (1998) 226-239.

[52] L.I. Kuncheva, A theoretical study on six classifier fusion strategies IEEE Trans. on Pattern Analysis and Machine Intelligence 24:2 (2002) 281-286.

[53] I. Mporas, T. Ganchev, N. Fakotakis, Speech segmentation using regression fusion of boundary predictions, Computer Speech and Language 24:2 (2010) 273-288.

[54] H.I. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, second ed., Morgan Kauffman Publishing, 2005.

[55] H. Akaike, A new look at the statistical model identification. IEEE Trans. on Automatic Control 19:6 (1974) 716-723.

[56] R. Kohavi, G.H. John, Wrappers for feature subset selection, Artificial Intelligence 97:1-2 (1997) 273-324.

[57] R.J. Quinlan, Learning with continuous classes, in Proc. of 5th Australian Joint Conference on Artificial Intelligence, Singapore (1992) 343-348.

[58] Y. Wang, I.H. Witten, Induction of model trees for predicting continuous classes, in poster papers of the 9th European Conference on Machine Learning, 1997.

[59] L. Breiman, Bagging predictors, Machine Learning 24:2 (1996)123-140.

[60] M. Kaariainen, T. Malinen, Selective rademacher penalization and reduced error pruning of decision trees, Journal of Machine Learning Research 5 (2004) 1107-1126.

[61] R. Vilalta, Y. Drissi, A perspective view and survey of meta-learning, Artificial Intelligence Review 18:2 (2002) 77-95.

[62] J. Platt, Fast training of support vector machines using sequential minimal optimization, in: B. Scholkopf, C. Burges, A. Smola (Eds.), Advances in kernel methods: Support vector learning, Cambridge, MA: MIT Press, 1999.

[63] A.J. Smola, B. Scholkopf, A tutorial on support vector regression, Royal Holloway College, London, U.K., NeuroCOLT Tech. Rep. TR 1998-030, 1998.

[64] B. Scholkopf, A.J. Smola, Learning with Kernels, MIT Press, 2002.

[65] J. Park, I.W. Sandberg, Approximation and radial-basis-function networks, Neural Computation, The MIT Press 5 (1993) 305-316.

[66] D. Aha, D. Kibler, Instance-based learning algorithms, Machine Learning 6 (1991) 37-66.

[67] J.A. Hartigan, M.A. Wong, A k-means clustering algorithm, Algorithm AS136, Applied Statistics 28 (1979) 100-108.

[68] CSTR US KED TIMIT. University of Edinburgh, 2001, http://www.festvox.org/dbs/dbs_kdt.html.

[69] P. Zervas, N. Fakotakis, G. Kokkinakis, Development and evaluation of a prosodic database for Greek speech synthesis and research, Journal of Quantitative Linguistics 15:2 (2008) 154-184.

[70] K. Silverman, M. Beckman, J. Pitrelli, M. Ostendorf, C. Wightman, P. Price, J. Pierrehumbert, J. Hirschberg, ToBI: A standard for labeling English Prosody, in Proc. of ICSLP-1992, Banff, Alberta, Canada, (1992) 867-870.

[71] D. Freedman, R. Pisani, R. Purves, Statistics, fourth ed., W.W. Norton & Company, New York London, 2007.

[72] L. Wang, Y. Zhao, M. Chu, J. Zhou, Z. Cao, Refining segmental boundaries for TTS database using fine contextual-dependent boundary models, in Proc. of ICASSP-04 (2004) 641-644.

[73] V. Vapnik, The Nature of Statistical Learning Theory, Springer, N.Y., 1995.

[74] V. Vapnik, Statistical Learning Theory, New York: Wiley, 1998.

[75] F. Wilcoxon, Individual comparisons by ranking methods, Biometric Bull 1 (1945) 80-83.

[76] K. Kira, L.A. Rendell, A practical approach to feature selection, in Proc. of 9th International Conference on Machine Learning, San Mateo, CA: Morgan Kaufmann (1992) 249-256.

[77] S.H. Hamraz, S.S. Feyzabadi, General-Purpose Learning Machine Using K-Nearest Neighbors Algorithm, RoboCup, volume 4020 of Lecture Notes in Computer Science, Springer, (2005) 529-536.

[78] Z. Liu, J. Yan, D. Zhang, and Q.L. Li, Automated tongue segmentation in hyperspectral images for medicine, Applied Optics 46 (2007) 8328-8334.

[79] G.E. Hughes, On the mean accuracy of statistical pattern recognizers, IEEE Trans. on Information Theory 14, (1968) 55-63.

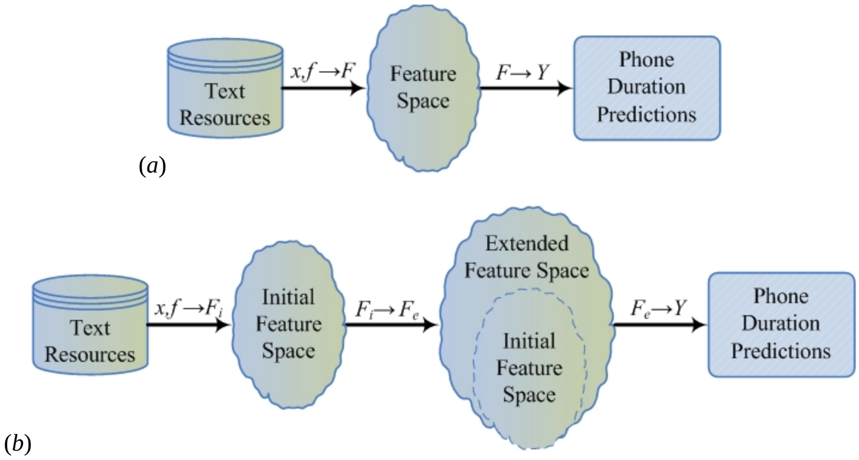

Fig. 1. Phone duration prediction: (a) the classical approach, (b) two-stage approach involving an intermediate feature vector extension step

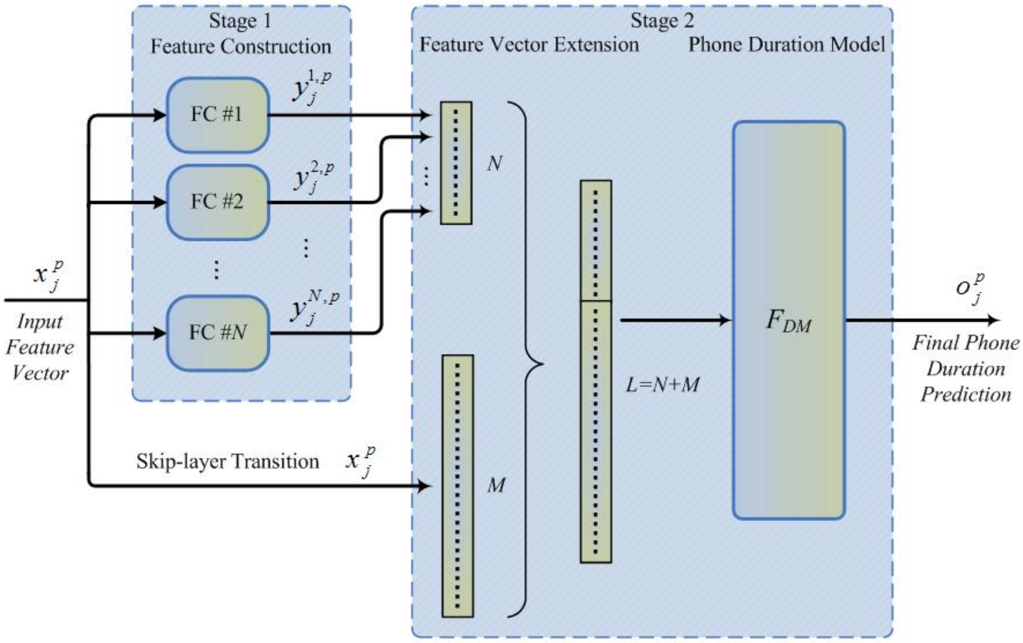

Fig. 2. Block diagram of the two-stage phone duration modelling scheme, which involves feature construction and feature vector extension

Table 1. Phone duration modelling methods studied in the literature.

Method	***Study***	***Database***	***Error Metric (ms)***	***Vowels***	***Consonants***
LR	[4]	American English male	RMSE	25.2	21.4
		Japanese male	RMSE	16.6	14.9
		Japanese female	RMSE	14.3	15.4
	[7]	RP English male	Standard deviation	20.3	
	[18]	Modern Greek female	RMSE	25.5	
	[25]	American English male	RMSE	22.89	
		Modern Greek female	RMSE	26.19	
CART	[4]	American English male	RMSE	26.4	21.5
		Japanese male	RMSE	17.2	14.2
		Japanese female	RMSE	14.9	13.9
	[7]	RP English male	Standard deviation	19.6	
	[8]	RP English male	RMSE	23.0	20.0
		RP English female	RMSE	26.0	21.0
		American English male	RMSE	27.0	24.0
	[9]	Indian Hindi	RMSE	27.1	
		Indian Telugu	RMSE	22.7	
	[11]	Korean	RMSE	22.0	
	[18]	Modern Greek female	RMSE	26.0	
	[22]	Korean	RMSE	27.5	24.2
	[25]	American English male	RMSE	22.72	
		Modern Greek female	RMSE	27.71	
Model Trees	[4]	American English male	RMSE	25.4	21.1
		Japanese male	RMSE	16.5	13.6
		Japanese female	RMSE	14.3	12.9
	[7]	RP English male	Standard deviation	19.4	
	[25]	American English male	RMSE	22.23	
		Modern Greek female	RMSE	27.17	
BN	[8]	RP English male	RMSE	1.5	4.1
		RP English female	RMSE	1.5	3.5
		American English male	RMSE	1.7	3.6
FFNN	[19]	European Portuguese	Standard deviation	19.5	
IBK	[18]	Modern Greek female	RMSE	27.5	
SoP	[8]	RP English male	RMSE	28.0	26.0
		RP English female	RMSE	25.0	25.0
		American English male	RMSE	32.0	33.0
	[22]	Korean	RMSE	32.1	28.9
Add. Regr. Trees	[4]	American English male	RMSE	24.5	20.2
		Japanese male	RMSE	16.1	12.8
		Japanese female	RMSE	13.9	12.1
	[25]	American English male	RMSE	21.35	
		Modern Greek female	RMSE	26.38	
Bagg. Trees	[4]	American English male	RMSE	25.8	20.9
		Japanese male	RMSE	16.7	13.9
		Japanese female	RMSE	14.5	13.5
	[25]	American English male	RMSE	22.14	
		Modern Greek female	RMSE	26.72	
SVR	[25]	American English male	RMSE	20.56	
		Modern Greek female	RMSE	25.21	
Fusion Scheme	[25]	American English male	RMSE	20.14	
		Modern Greek female	RMSE	24.76	

Table 2. Mean Absolute Error (MAE), standard deviation of absolute error (STD of AE) and Root Mean Square Error (RMSE) in milliseconds for the eight feature constructors (FCs) on: (*a*) the KED TIMIT database, and (*b*) the WCL-1 database. Bold fonts indicate the best model among the eight FCs.

(*a*) results on the KED TIMIT database

FC algorithms	***MAE (ms)***	***STD of AE (ms)***	***RMSE (ms)***
SMOreg	**14.95**	**14.11**	**20.56**
Add. Regr. m5pR	15.82	14.34	21.35
Add. Regr.REPTrees	16.29	15.06	22.19
Bagg. m5pR	16.51	14.76	22.14
m5p	16.62	14.77	22.23
Bagg. REPTrees	16.69	15.89	23.04
m5pR	16.93	15.16	22.72
LR	17.15	15.16	22.89

(*b*) results on the WCL-1 database

FC algorithms	***MAE (ms)***	***STD of AE (ms)***	***RMSE (ms)***
SMOreg	**16.78**	**18.81**	**25.21**
LR	18.00	19.02	26.19
Add. Regr.REPTrees	18.08	19.97	26.94
Add. Regr.m5pR	18.13	19.16	26.38
Bagg. m5pR	18.14	19.63	26.72
m5p	18.31	20.08	27.17
Bagg. REPTrees	18.93	20.32	27.77
m5pR	19.07	20.10	27.71

Table 3. Root Mean Square Error in milliseconds per phonetic category for the eight feature constructors (FCs) on: (*a*) the KED TIMIT database, and (*b*) the WCL-1 database. Bold fonts indicate the best model for each phonetic category among the eight FCs.

(*a*) results on the KED TIMIT database

Clustering	***LR***	***m5p***	***m5pR***	***Add. Reg.***		***Bagg.***		***SMOreg***
				m5pR	***REPTrees***	***m5pR***	***REPTrees***	
Vowel	24.56	24.18	25.46	23.67	24.87	24.78	26.34	**22.72**
Consonant	21.72	20.86	20.74	19.69	20.24	20.24	20.60	**19.02**
Affricate	22.44	24.41	23.48	22.86	**21.72**	22.96	23.34	21.88
Approximant	22.23	22.44	23.09	21.77	22.56	22.59	24.07	**20.42**
Fricative	22.51	21.67	21.10	20.19	20.69	20.63	20.96	**19.63**
Lateral	21.16	20.98	21.18	20.29	21.16	20.52	21.89	**19.77**
Nasal	18.59	17.88	17.80	17.11	16.94	17.28	17.57	**16.53**
Stop	23.39	22.07	21.62	20.26	20.97	21.04	20.80	**19.61**

(*b*) results on the WCL-1 database

Clustering	***LR***	***m5p***	***m5pR***	***Add. Regr.***		***Bagg.***		***SMOreg***
				m5pR	***REPTrees***	***m5pR***	***REPTrees***	
Vowel	24.22	24.68	26.04	24.51	25.18	24.91	26.62	**23.12**
Consonant	27.86	29.25	29.13	27.97	28.44	28.27	28.77	**26.57**
Fricative	25.67	27.04	26.93	25.79	26.23	25.57	26.45	**23.95**
Liquid	19.46	19.19	19.55	18.84	17.83	18.47	18.02	**16.38**
Nasal	22.44	22.94	23.11	22.27	22.15	22.18	22.27	**20.62**
Stop	34.09	36.22	35.88	34.46	35.46	35.35	35.98	**33.53**

Table 4 (*a*). Root Mean Square Error in milliseconds per phone for the eight feature constructors (FCs) on the KED TIMIT database. Bold fonts indicate the best model for each phone among the eight FCs.

Phones	***LR***	***m5p***	***m5pR***	***Add. Regr.***		***Bagg.***		***SMOreg***
				m5pR	***REPTrees***	***m5pR***	***REPTrees***	
aa	27.81	25.57	28.01	**24.57**	27.27	26.71	29.22	25.64
ae	31.40	30.97	31.67	29.75	31.19	31.64	33.13	**29.23**
ah	19.67	22.34	22.27	20.31	21.50	20.88	22.27	**19.38**
ao	32.79	**29.21**	32.95	30.54	32.11	32.66	33.29	29.65
aw	33.46	**32.89**	37.35	34.55	38.49	37.25	40.02	33.07
ax	16.10	15.66	16.06	15.16	15.56	15.54	15.93	**14.80**
ay	37.12	**32.78**	38.37	34.43	34.04	36.64	37.23	34.51
b	23.89	22.36	23.42	22.24	23.03	22.52	**21.19**	21.33
ch	**19.69**	23.34	21.17	20.48	20.43	19.89	22.36	20.57
d	20.77	19.36	19.66	19.12	20.05	19.32	20.54	**18.26**
dh	17.56	16.03	15.72	15.19	**14.57**	15.16	15.30	15.14
dx	11.08	10.38	11.30	9.99	**8.78**	9.54	8.86	9.63
eh	20.94	20.39	22.50	21.41	21.44	21.32	22.61	**19.05**
el	21.39	27.24	21.52	20.79	**18.98**	19.97	21.05	22.32
em	13.61	15.31	10.51	10.44	**10.13**	10.28	13.99	11.58
en	22.26	24.60	25.01	23.18	**20.67**	22.44	21.80	21.01
er	28.41	29.28	28.73	27.09	27.77	28.15	29.87	**25.29**
ey	27.76	26.99	29.43	28.12	29.90	28.72	31.36	**26.73**
f	22.84	23.90	21.52	20.09	21.05	21.08	22.43	**18.91**
g	18.23	17.14	18.73	17.04	17.65	17.88	17.62	**16.22**
hh	19.13	18.79	18.82	18.52	18.73	18.28	18.73	**17.54**
ih	19.38	19.76	20.16	19.09	19.81	19.82	20.86	**17.53**
iy	23.04	23.06	23.87	22.05	24.93	23.27	25.39	**20.99**
jh	24.36	25.22	25.14	24.56	**22.68**	25.08	24.07	22.85
k	22.18	21.82	20.62	18.65	18.63	19.94	18.93	**17.64**
l	21.13	20.18	21.14	20.24	21.39	20.58	21.98	**19.47**
m	16.07	15.32	16.20	15.45	16.19	15.81	17.04	**14.38**
n	18.69	17.65	17.29	16.70	**16.18**	16.80	16.32	16.19
ng	22.38	20.86	20.13	**19.90**	20.88	20.61	22.41	20.91
ow	28.12	28.98	28.93	27.20	28.85	27.73	30.68	**25.54**
oy	**25.45**	30.16	34.58	28.81	30.61	33.13	34.72	31.19
p	25.06	24.90	22.50	21.05	21.32	21.94	21.25	**20.45**
r	19.20	18.84	20.18	19.28	20.11	19.92	21.18	**18.25**
s	26.37	24.47	24.31	23.46	24.45	24.36	24.54	**23.21**
sh	19.71	21.72	19.28	18.30	20.53	18.49	20.27	**16.41**
t	28.18	25.60	25.06	23.64	25.14	24.72	24.93	**23.37**
th	24.09	26.39	29.14	25.58	**21.31**	25.21	22.59	22.05
uh	20.64	20.61	23.10	20.45	25.35	22.68	26.16	**19.88**
uw	27.65	27.73	29.05	28.00	30.35	29.40	33.64	**24.97**
v	17.26	17.31	16.72	16.93	17.15	16.66	17.34	**16.26**
w	20.28	20.09	22.35	19.81	20.93	20.89	22.59	**19.12**
y	18.36	19.08	18.85	18.80	19.42	19.22	20.56	**16.34**
z	22.38	20.42	19.94	19.07	19.37	19.24	19.10	**18.99**
zh	25.60	28.40	25.25	**22.62**	26.38	23.95	27.28	24.66

Table 4 (*b*). Root Mean Square Error in milliseconds per phone for the eight feature constructors (FCs) on the WCL-1 database. Bold fonts indicate the best model for each model among the eight FCs.

Phones	***LR***	***m5p***	***m5pR***	***Add. Regr.***		***Bagg.***		***SMOreg***
				m5pR	***REPTrees***	***m5pR***	***REPTrees***	
a	24.25	25.85	25.76	24.07	24.57	24.83	26.07	**22.71**
b	21.05	24.66	24.41	21.84	22.05	22.33	22.53	**20.20**
c	24.16	28.40	25.43	22.48	**20.29**	26.62	23.62	20.85
D	**22.64**	23.08	25.08	24.24	24.39	24.00	26.25	**22.64**
d	**19.33**	20.4	23.54	21.01	21.44	21.39	24.61	20.10
e	25.05	25.11	26.69	25.62	26.79	25.71	26.48	**24.05**
f	30.13	34.11	30.41	29.95	33.13	**29.50**	31.94	29.56
G	30.72	37.75	37.14	31.68	31.56	33.82	33.99	**29.89**
g	34.43	38.79	34.94	35.14	40.05	34.30	37.80	**33.85**
h	24.73	25.91	26.39	24.69	24.87	23.78	25.88	**23.50**
i	24.17	24.30	25.54	24.27	24.68	24.68	26.98	**23.09**
j	25.65	26.18	23.13	21.75	**20.52**	24.39	23.48	21.50
K	45.75	44.50	45.47	43.94	46.86	45.73	45.82	**43.28**
k	**42.27**	46.15	44.65	43.31	44.58	43.90	47.61	43.61
ks	**22.50**	24.00	42.80	39.97	26.32	42.34	27.10	23.16
l	19.95	19.34	20.63	19.85	20.17	19.65	20.96	**18.31**
L	**24.98**	32.86	32.93	29.20	28.64	29.98	29.43	26.64
m	22.90	22.82	23.74	22.56	23.46	22.67	23.96	22.27
N	26.99	33.30	36.39	33.22	24.13	34.11	24.37	**21.26**
n	21.76	22.14	21.58	21.21	21.07	20.96	20.88	**19.38**
o	23.70	23.81	25.99	24.18	25.08	24.36	25.85	**22.72**
p	29.65	32.71	31.57	**28.65**	30.51	29.80	30.04	28.24
Q	**23.08**	25.82	25.22	23.49	24.99	23.82	26.83	23.85
r	18.64	17.53	17.30	17.06	14.94	16.41	14.53	**13.85**
s	26.93	27.65	27.28	26.10	24.75	25.47	25.11	**23.49**
t	34.70	36.84	34.98	34.31	36.09	34.96	36.04	**34.07**
u	23.24	**22.63**	27.51	24.78	25.11	25.37	29.45	23.12
v	**23.87**	24.11	26.09	25.80	34.70	25.56	27.08	24.86
w	**20.83**	25.71	40.93	42.47	25.92	42.98	29.62	23.66
X	22.75	24.38	26.33	24.45	23.44	25.03	25.95	**21.44**
x	**20.33**	24.82	26.45	23.35	21.87	24.98	25.06	21.58
y	20.77	20.35	22.98	21.03	21.35	21.38	21.64	**19.68**
Y	**26.68**	28.56	29.65	28.08	27.41	28.37	28.39	26.82
z	23.05	22.38	23.31	22.64	23.13	22.98	24.68	21.58

Table 5. Mean Absolute Error (MAE), standard deviation of absolute error (STD of AE) and Root Mean Square Error (RMSE) in milliseconds for the baseline FCF scheme and for the proposed FVE scheme on: (*a*) the KED TIMIT database, and (*b*) the WCL-1 database. Bold fonts indicate the best model among the ten models.

(*a*) results on the KED TIMIT database

PDM algorithms	**baseline, FCF**			**proposed scheme, FVE**		
	MAE	***STD of AE***	***RMSE***	***MAE***	***STD of AE***	***RMSE***
Add. Regr. m5pR	15.72	14.94	21.69	15.74	14.92	21.69
Add. Regr. REPTrees	15.79	14.94	21.74	15.60	14.76	21.47
Bagg. m5pR	15.81	15.09	21.86	15.83	15.08	21.86
Bagg. REPTrees	15.88	15.15	21.95	16.26	15.52	22.48
IBK	15.19	14.69	21.02	17.41	15.88	23.57
LR	15.49	14.45	21.18	15.40	14.40	21.08
m5p	15.56	14.60	21.34	15.45	14.48	21.17
m5pR	15.97	15.28	22.10	15.95	15.25	22.06
RBFNN	15.53	14.49	21.24	21.28	18.37	28.11
SMOreg	14.66	13.82	20.14	**14.36**	**13.56**	**19.75**
best FC (SMOreg)	–	–	–	14.95	14.11	20.56

(*b*) results on the WCL-1 database

PDM algorithms	**baseline, FCF**			**proposed scheme, FVE**		
	MAE	***STD of AE***	***RMSE***	***MAE***	***STD of AE***	***RMSE***
Add. Regr. m5pR	17.69	19.84	26.58	17.75	19.93	26.69
Add. Regr. REPTrees	18.00	20.56	27.32	17.89	20.22	26.99
Bagg. m5pR	17.72	19.84	26.60	17.75	19.82	26.60
Bagg. REPTrees	17.99	20.45	27.23	18.15	20.51	27.39
IBK	16.98	18.85	25.47	18.53	20.29	27.48
LR	18.32	20.19	27.26	18.22	20.11	27.14
m5p	17.84	20.51	27.18	17.81	20.46	27.12
m5pR	17.91	20.00	26.85	17.98	20.02	26.91
RBFNN	17.34	19.51	26.10	21.29	21.43	30.21
SMOreg	16.35	18.59	24.76	**15.97**	**18.34**	**24.04**
best FC (SMOreg)	–	–	–	16.78	18.81	25.21

Table 6. Mean Absolute Error (MAE), standard deviation of absolute error (STD of AE) and Root Mean Square Error (RMSE) in milliseconds for the various feature subsets on the PDMs on: (*a*) the KED TIMIT database, and (*b*) the WCL-1 database. Bold fonts indicate the best model among the ten models.

(*a*) results on the KED TIMIT database

PDM algorithms	**PDM-Set 20**			**PDM-Set 50**			**PDM-Set 80**			**PDM-Full set**		
	MAE	***STD of AE***	***RMSE***	***MAE***	***STD of AE***	***RMSE***	***MAE***	***STD of AE***	***RMSE***	***MAE***	***STD of AE***	***RMSE***
Add. Regr. m5pR	15.73	14.93	21.69	15.7	14.90	21.64	15.71	14.90	21.65	15.74	14.92	21.69
Add. Regr. REPTrees	15.63	14.80	21.53	15.66	14.79	21.54	15.63	14.76	21.49	15.60	14.76	21.47
Bagg. m5pR	15.81	15.08	21.85	15.80	15.08	21.84	15.82	15.08	21.86	15.83	15.08	21.86
Bagg. REPTrees	16.26	15.52	22.48	16.27	15.51	22.48	16.26	15.50	22.46	16.26	15.52	22.48
IBK	15.78	14.71	21.57	16.81	15.85	23.10	17.11	15.78	23.28	17.41	15.88	23.57
LR	15.40	14.42	21.09	15.39	14.41	21.08	15.39	14.40	21.07	15.40	14.40	21.08
m5p	15.48	14.47	21.18	15.50	14.63	21.31	15.46	14.45	21.16	15.45	14.48	21.17
m5pR	15.96	15.28	22.10	15.96	15.26	22.08	15.95	15.27	22.08	15.95	15.25	22.06
RBFNN	16.51	14.81	22.18	18.85	17.01	25.39	20.38	17.83	27.08	21.28	18.37	28.11
SMOreg	14.54	13.75	20.01	14.44	13.66	19.87	14.37	13.60	19.79	**14.36**	**13.56**	**19.75**
best FC (SMOreg)	–	–	–	–	–	–	–	–	–	14.95	14.11	20.56

(*b*) results on the WCL-1 database

PDM algorithms	**PDM-Set 20**			**PDM-Set 50**			**PDM-Set 80**			**PDM-Full set**		
	MAE	***STD of AE***	***RMSE***	***MAE***	***STD of AE***	***RMSE***	***MAE***	***STD of AE***	***RMSE***	***MAE***	***STD of AE***	***RMSE***
Add. Regr. m5pR	17.76	19.92	26.69	17.76	19.98	26.74	17.74	19.89	26.65	17.75	19.93	26.69
Add. Regr. REPTrees	17.93	20.57	27.28	17.93	20.41	27.17	17.96	20.57	27.31	17.89	20.22	26.99
Bagg. m5pR	17.73	19.83	26.60	17.74	19.84	26.62	17.75	19.83	26.61	17.75	19.82	26.60
Bagg. REPTrees	18.13	20.53	27.39	18.14	20.54	27.40	18.14	20.54	27.41	18.15	20.51	27.39
IBK	17.24	19.32	25.90	17.96	20.05	26.92	18.19	20.20	27.18	18.53	20.29	27.48
LR	18.20	20.11	27.12	18.21	20.07	27.10	18.23	20.09	27.13	18.22	20.11	27.14
m5p	17.89	20.47	27.18	17.83	20.22	26.96	17.78	20.07	26.81	17.81	20.46	27.12
m5pR	17.93	20.01	26.87	17.96	20.01	26.89	17.95	20.02	26.89	17.98	20.02	26.91
RBFNN	17.94	19.77	26.70	19.81	20.76	28.69	20.69	20.99	29.47	21.29	21.43	30.21
SMOreg	16.28	18.62	24.68	16.17	18.57	24.39	16.05	18.65	24.18	**15.97**	**18.34**	**24.04**
best FC (SMOreg)	–	–	–	–	–	–	–	–	–	16.78	18.81	25.21