Time-Series Classification based on Individualised Error Prediction

Krisztian Buza, Alexandros Nanopoulos, Lars Schmidt-Thieme
Information Systems and Machine Learning Lab
University of Hildesheim
Hildesheim, Germany
$\{$*buza,nanopoulos,schmidt-thieme*$\}$*@ismll.de*

***Abstract*—Time-series classification is an active research topic in machine learning, as it finds applications in numerous domains. The k-NN classifier, based on the discrete time warping (DTW) distance, had been shown to be competitive to many state-of-the art time-series classification methods. Nevertheless, due to the complexity of time-series data sets, our investigation demonstrates that a single, global choice for k (≥ 1) can become suboptimal, because each individual region of a data set may require a different k value. In this paper, we proposed a novel individualized error prediction (IEP) mechanism that considers a range of k-NN classifiers (for different k values) and uses secondary regression models that predict the error of each such classifier. This permits to perform k-NN time-series classification in a more fine grained fashion that adapts to the varying characteristics among different regions by avoiding the restriction of a single value of k. Our experimental evaluation, using a large collection of real time-series data, indicates that the proposed method is more robust and compares favorably against two examined baselines by resulting in significant reduction in the classification error.**

***Keywords*-time series; classification; error estimation**

I. Introduction

Time-series classification is an active research topic in machine learning and data mining, because it finds several applications including finance, medicine, biometrics, chemistry, astronomy, robotics, networking and industry [11]. The increasing interest in time-series classification resulted in a plethora of different approaches ranging from neural [16] and Bayesian networks [22] to genetic algorithms, support vector machines [6] and frequent pattern mining [7], [1]. Nevertheless, despite its simplicity, the 1-nearest neighbor (1-NN) classifier based on the dynamic time warping (DTW) distance [21] has been shown to be competitive, if not superior, to many state-of-the art time-series classification methods [20], [2], [13]. Due to its good performance, this method has been examined in depth (a thorough summary of results can be found at [18]) with the aim to improve its accuracy [19] and efficiency [12].

The wide acceptance of 1-NN in time-series classification has been supported also by its lack of parameters. Nevertheless, it is known that the choice of parameter k in the k-NN classifier affects the bias-variance trade-off [9]. Smaller values of k may result in increased variance due to overfitting, whereas larger values of k increase the bias by capturing only global tendencies. Recent studies [17] have indicated that, due to intrinsic characteristics in time-series data sets, such as the mixture between the different classes, the dimensionality, and the skewness in the distribution of error (i.e., the existence of "bad hubs" [17] that account for a surprisingly large fraction of the total error), significant improvement in the accuracy of the k-NN time-series classification can be attained with k being larger than 1. In such cases, k can be determined using a hold-out subset of the training data.

Since time-series data sets tend to be complex, their intrinsic characteristics, such as those mentioned above, may vary over different regions. As a consequence, setting a single, global choice for k (≥ 1) can become suboptimal, since each individual region of a data set may require a different value of k. Therefore, the motivation in our study is to investigate how to perform k-NN time-series classification in a more fine grained fashion that adapts to the varying characteristics among different regions by avoiding the restriction of a single value of k.

In this paper, we propose a mechanism that considers a range of values for k and estimates for each time-series, t, that has to be classified, the likelihood of erroneous classification (henceforth called simply *error*) of t by the k-NN classifier. The estimation is computed for each k in the examined range, in order to select the one that minimizes the error for t. Since we propose the classification of time-series by predicting the error individually for each of them, the proposed approach is called time-series classification based on *individualized error prediction* (IEP). IEP is performed by regression models that are trained in order to make accurate estimations for the error of the k-NN classifier. In summary, our contribution are described as follows:

1) We introduce the notion of IEP, which can be generally applied in classification problems (i.e., not just for the k-NN classification of time-series).
2) We propose a novel mechanism for IEP, which is applied to the task of classifying time-series data based on the k-NN classifier.
3) We perform a thorough experimental evaluation, with a large number of commonly used benchmark data sets, which indicates significant improvement in accuracy attained by the proposed method when compared with the widely used 1-NN classifier and with the

k-NN classifier that determines a single optimal k ($k \geq 1$). The attained improvement happens especially when the characteristics of the data sets become challenging for these two baselines.

The rest of this paper is organised as follows: in Section II we overview the related work, in Section III we outline IEP, whereas in Section IV we describe the proposed algorithm. In Section V we present our experimental evaluation. We provide our conclusions in Section VI.

II. Related Work

Despite the development of a large variety of models for time-series classification [16], [6], [7], the k-nearest neighbor (k-NN) classifier (especially for $k = 1$), has been shown to be competitive to many other, more complex models [20], [2], [13]. Nearest-neighbor classification of time series uses Dynamic Time Warping (DTW) [21], because it is an elastic distance measure, i.e., it is robust w.r.t. shiftings and elongation in the time series. Recent works aimed at making DTW more accurate and scalable [19], [12]. DTW was analyzed from a theoretical-empirical point of view in [18], whereas Ding et al. found no other distance measure that significantly outperforms DTW [2].

Our proposed approach, i.e., using IEP to select the best k for k-NN time-series classification, could be related to works that perform local adaptation of k-NN classifier. A locally adaptive distance measure was proposed by Hastie and Tibshirani [8], while Domeniconi and Gunopulos [3] used SVMs to define a local measure of feature relevance, i.e., feature weights depending on the location of a data point to be classified. In [4] adaptive nearest neighbor classification in high-dimensional spaces was studied. In contrast to these works, our IEP approach adapts by selecting the proper value of k and not by determining a localized distance function.

Ougiaroglou at al. [15] presented 3 early-break heuristics for k-NN which can be interpreted as adapting the number of nearest neighbors. Their heuristics, however, aimed at speeding-up k-NN, while we focus on making nearest neighbor classification more accurate using the principled framework of IEP.

Error-prediction methods are usually applied globally in order to estimate the overall performance of a classification model [14], [10]. In our approach, we focus on individualised error prediction. This is similar to learning the residuals, i.e., the difference between predicted and actual labels. Duffy and Helmbold followed this direction and incorporated residuals into boosting of regression models [5]. In contrast to this work, we do not focus on boosting. Similarly to our work, Tsuda et al. [23] proposed an individualised approach for predicting the leave-one-out error of vector classification with support vector machines (SVM) and linear programming machines (LPM). Compared to this work, our proposed approach performs general individualised error prediction (not just for leave one out). More importantly our approach *exploits* error prediction to improve accuracy of classification and not as a *per se* task, as done in [23].

A set of earlier approaches to localized error prediction for the k-NN classifier was proposed by Wettschereck and Dietterich [24]. However, these approaches were based solely on heuristics such as using different k values per class or per cluster (after clustering the training set). Our proposed framework is principled and more generic than these simple approaches: we distinguish between the error prediction step and classification step, our framework supports systematic usage of the predicted error-likelihoods, and our framework allows various classification and regression models.

Finally, all aforementioned works concerned with classification of vectors (point data), while we focus on time-series classification.

III. Individualized Error Prediction

In this section, we introduce the concept IEP, which is the basis of the proposed algorithm that will be detailed in Section IV. We first provide a motivating example and then outline the approach we take for IEP.

A. Motivating example

As mentioned in Section I, the selection of a single value of k for the k-NN time-series classification, can lead to suboptimal accuracy, because of varying characteristics among different regions of the data. We investigate this phenomenon in more detail by first presenting a motivating example for the simple setting of binary classification of a 2-dimensional data set.[1]

Figure 1 depicts a set of labeled instances from two classes that are denoted by triangles and circles. The density in the class of triangles (upper region) is larger than in the class of circles (lower region). We consider two test instances, denoted as ‘1’ and ‘2’, that have to be classified. We also assume that the ground-truth considers test instance ‘1’ as a triangle, whereas ‘2’ as a circle. For ‘1’, its 1-NN is a circle. Thus, the 1-NN method classifies ‘1’ incorrectly. Using the k-NN classifier with $k > 1$ (e.g., in the range between 3 and 6), we can overcome this problem. However, the selection of a single k from the above range results in incorrect classification of test instance ‘2’. Due to the lower density in the circles’ class, by setting $k > 1$ (e.g., inside the above range), we detect neighbors of ‘2’ whose majority belongs to the triangles’ class (we assumed ‘2’ is a circle). This can be observed in Figure 1, where the large dashed cycle around ‘2’ shows that among all its 6-NN, only 1 belongs to the circles’ class. Thus, unlike for ‘1’, $k = 1$ is a good choice for ‘2’, because its 1-NN (shown inside the smaller dashed cycle) has the correct class.

[1] In this example, we use a 2-dimensional data set, thus we depart for the moment from the examination of time-series data that are in general high-dimensional, in order to ease the presentation with an illustrative figure.

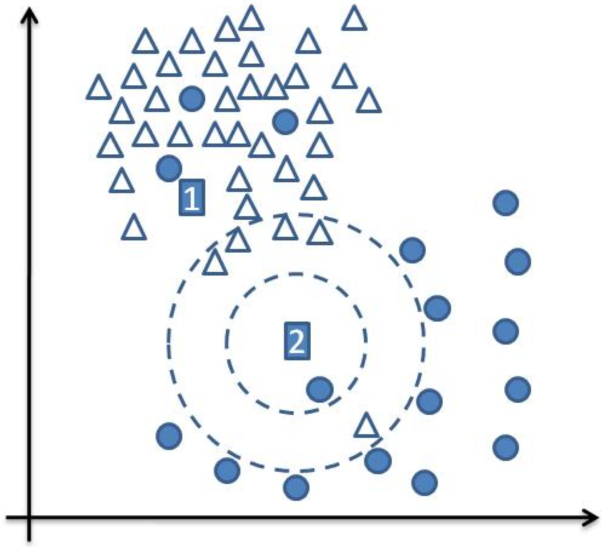

Figure 1. The optimal choice of the number of nearest neighbors is not unique for the entire data, but it may be different from region to region: in case of the classification of the unlabeled instance denoted by '1', $k > 1$ (e.g., $k = 3$) is required; whereas for '2' we should choose $k = 1$.

The exemplified problem is amplified with time-series data due to their higher dimensionality and complexity. We propose to choose the value of k on an individualized basis, i.e., separately for each test instance to be classified. Following this approach in the example of Figure 1, besides the k-NN classifier, we need an additional model, which will allow for predicting that $k_1 = 3$ is a good choice, when we classify instance '1', whereas $k_2 = 1$ is an appropriate choice for the classification of instance '2'. In the following we outline how the proposed approach can be developed.

B. Outline of IEP

We propose a mechanism for individualized prediction of the error with which the k-NN classifier will misclassify each test instance, t. This mechanism considers a range of k values and predicts the error of each corresponding k-NN classifier.[2] Error prediction is performed by introducing secondary regression models that are trained to predict the error of each considered k-NN classifier.

The examined range of n values for k is denoted as $\{k_i\}_{i=1}^{n}$. $E(t, k_i)$ denotes the error, i.e., the likelihood with which the k_i-NN classifier ($1 \leq i \leq n$) will misclassify t. We select k^* that minimizes the predicted error: $k^* = \text{argmin}_{k_i, 1 \leq i \leq n}\{E(t, k_i)\}$. Finally, the k^*-NN classifier is used to classify t. This is shown in Fig. 2.

[2] Although this range is user-defined, its determination is much simpler and intuitive compared to selecting a single k. This will be asserted by our experimental results, which indicate that the range $1 - 10$ was appropriate for all examined benchmark data sets.

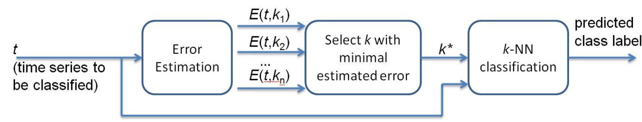

Figure 2. Summary of IEP for k-NN classification.

The previous description provides only the outline of IEP. A concrete algorithm for times-series classification is described in the following, by specifying also the secondary models that perform error prediction.

IV. Time-series classification based on IEP

The proposed mechanism for IEP-based classification involves two types of models:

- Primary models, which classify time series with the k-NN classifier (based on the DTW distance).
- Secondary models, which predict the error of the primary models.

To train the secondary models, we partition the original training data set, D, in two disjoint subsets D_1 and D_2 (i.e., $D_1 \cup D_2 = D, D_1 \cap D_2 = \emptyset$). D_2 is called hold-out set. For each time series $t \in D_2$, and for each examined value of k_i in a range $\{k_i\}_{i=1}^{n}$, we use D_1 to classify t with the k_i-NN classifier. Based on the class label of t that is given in D_2, we determine if the k_i-NN classifier (for each $1 \leq i \leq n$) has correctly classified t. In case of correct classification, we associate with t an error value of 0, otherwise we associate an error value of 1. Thus, from the hold-out set D_2 we can generate n new data sets D'_i, $1 \leq i \leq n$. Each D'_i contains all time-series of the hold-out set D_2 along with their associated error values (0 or 1) for the corresponding k_i-NN classifier.

Next, each generated D'_i acts as the training set for the corresponding secondary model. Thus, based on the associated error values in each D'_i, the corresponding secondary model is trained as a regression model in order to predict the error of the k_i-NN classifier (i.e., the corresponding primary model). This procedure is summarized in Figure 3.

We implement each secondary level model as a k'-NN *regression model* based on the DTW distance. (We denote k' in order to distinguish from k that is used in the primary k-NN classification models.) The secondary level prediction for a time series t is calculated the following way:

$$E(t, k_i) = \frac{\sum_{t_N \in \mathcal{N}(t)} a_e(t_N, k_i)}{k'}$$

where $\mathcal{N}(t) \subset D_2$ is the set of k'-NNs of t and $a_e(t_N, k_i)$ is the associated error value of each $t_N \in \mathcal{N}(t)$.

Similar to the primary models, the secondary models also have to determine the value of k'. Nevertheless, the proposed approach has the advantage that, as will be asserted by our

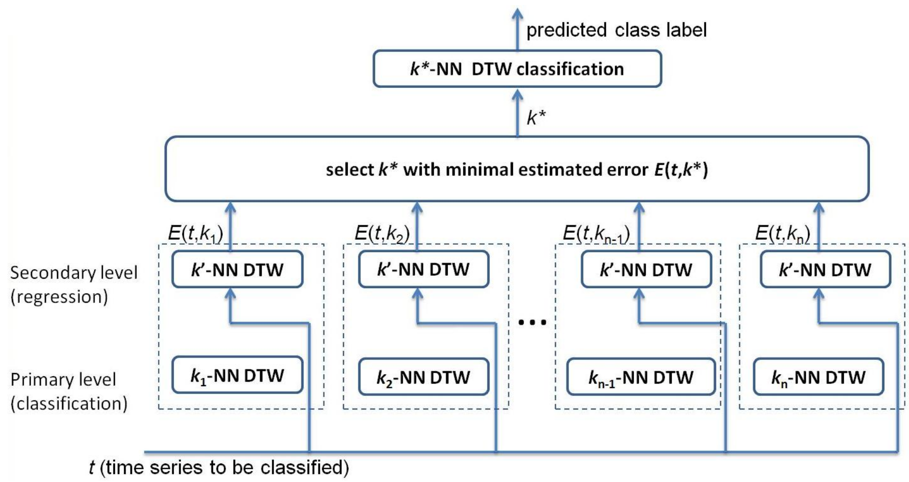

Figure 3. Schema of IEP-based time series classification.

experimental results, it suffices to specify a single value of k' that will be used by *all* secondary models. More importantly, the selection of k' is simple due to the small variance that can be inspected in the accuracy of the secondary models.[3] Thus, while for the primary models the best k varies among different regions inside a dataset, at the secondary level, due to the different learning task performed by the secondary models, we can observe a stable tendency regarding k'.

Finally, the classification of a time series is done by the primary model that is predicted (based on the IEP mechanism) to result in the smallest error for this time series. We have to clarify that the training of secondary models is being performed in an off-line fashion, i.e., using only the training sets that are generated by the hold-out set. Therefore, the use of the IEP mechanism produces a small overhead to the (online) time needed to classify a time series, as we need to probe the n trained secondary models to retrieve their estimated errors. Moreover, our experimental results we will demonstrate that also the overhead in the off-line time for the training of the secondary models, is manageable even for large time-series data sets.

V. Experimental Evaluation

A. Experimental configuration

To assist reproducibility, we provide a detailed description of the configuration of our experiments.

Methods. We compare the proposed method, denoted as IEP, against two baselines: the 1-NN classifier and the k-NN classifier that selects k using a hold-out set from the training data. The latter baseline uses the same hold-out set as the proposed method, examines the same range of values for k, and selects the one that produces the smallest average error for all time series in the hold-out set. All examined methods are based on the same DTW distance that constrains the warping window size at 5% around the matrix diagonal [18].

Data-sets. Out of all the 38 data sets used in [2], we examined 35 data sets: we excluded 3 of them (Coffee, Beef and OliveOil) due to their tiny size (less than 100 time series). The names of the remaining data sets and their size (number of time series they contain) are listed in the first and second columns of Table I.

Parameters. For the k-NN classifiers at the primary level of the proposed method, we examine all k values in the range $1-10$. We experimented with larger k values as well, but we observed that they increase the bias and deteriorate the resulting accuracy. For the k'-NN regression models at the secondary level of the proposed method, as mentioned in Section IV, it is sufficient to determine a single value of k' for all secondary models and for all examined data sets. In the following, we use the default value of $k' = 5$, but we also test the sensitivity against k'.

Comparison protocol. We measure the misclassification error using 10-fold cross validation, with the exception of three data sets (FaceFour, Lighting2, and Lighting7) for which we used the leave-one-out protocol due to their small size. In each round of the 10-fold cross validation, out of the 9 training splits, we used 5 to train the primary models (D_1), the rest 4 splits served as hold-out data (D_2).[4] For classifying test data, i.e., after selecting for IEP and k-NN the best k, we can again use all training splits.

After using the above evaluation procedure, we made a striking observation about the performance of all examined methods (proposed and baselines): in the majority of data sets, the misclassification error was rather low (less than 5%). To have a challenging comparison with non trivial classification, we choose to affect intrinsic characteristics

[3] In our experimental results, we found that $k' = 5$ was appropriate for all examined benchmark data sets.

[4] Ratios other than the examined 5-4, gave similar results. In case of leave-one-out, the training data was split according to 5 to 4 proportion.

of the data sets. According to the findings in [17], time-series data sets usually have high intrinsic dimensionality and, thus, some of their instances tend to misclassify a surprisingly large number of other instances when using the k-NN classifier ($k \geq 1$). These instances are called "bad hubs" and are responsible for a very large fraction of the total error. For this reason, for each time series, t, in a data set, we measured two quantities: the badness $B(t)$ of t and the goodness $G(t)$ of t. $B(t)$ ($G(t)$, resp.) is the total number of time series in the data set, which have t as their first nearest neighbor while having different (same, resp.) class label from t. For each data set, we sort all time series according to the $G(t) - B(t)$ quantity in descending order. Then we change the label of first p percent time series in this ranking (p varies in range 0-10%).[5] Since the above procedure results in data sets that have stronger "bad hubs" and a less clear separation between classes, the comparison among the examined methods becomes more challenging and can characterize better the robustness of the methods.

B. Experimental Results on Classification Accuracy

The results on classification error are summarized in Table I. For brevity, we only report results at $p = 1\%, 5\%$, and $p = 10\%$ noise, however we observed similar tendencies at all other noise ratios in the examined range of p. Bold font denotes the winning method for each data set. In case where winner is IEP, we also provide two symbols in the form: $\pm/\pm$ to denote the result of statistical-significance test (t-test at 0.05 level) against 1-NN and k-NN, respectively, where a $+$ denotes significance and $-$ its absence. In case where the winner is not IEP, we provide only the result (again in form of $\pm$) of statistical-significance test of the winner against IEP.

Table II summarizes these results by reporting the number of cases, per noise level and in total, that IEP wins/looses against 1-NN and k-NN (in parenthesis we report in how many cases wins/looses are statistically significant).

	$p = 1$ %	$p = 5$ %	$p = 10$ %	total
Wins against 1-NN	30 (20)	34 (29)	34 (31)	98 (80)
Looses against 1-NN	5 (1)	1 (0)	1 (0)	7 (1)
Wins against k-NN	30 (15)	30 (9)	28 (14)	88 (38)
Looses against k-NN	5 (1)	5 (1)	7 (1)	17 (3)

Table II
NUMBER IEP'S WINS/LOOSES AGAINST 1-NN AND k-NN.

As shown, in the vast majority of the cases IEP outperforms its competitors, often significantly, whereas when it looses, the difference is usually non-significant. In several cases, the error of IEP is an order of magnitude lower (e.g., for TwoLeadECG at $p = 1\%$, for GunPoint and Trace at $p = 5\%$, furthermore for Plane at $p = 10\%$).

C. Secondary model's quality

We also examined the quality of the secondary model's error prediction. To gain insight in the role of k', Figure 4 depicts the root mean squared error (RMSE) of the secondary models as function of k' for two characteristic datasets. Increasing k' leads to improvement of the secondary models first. However, after a point ($k' = 5$ in this cases) the quality of the secondary models becomes stable, i.e. it does not change significantly. This tendency is similar in the range $5 \leq k' \leq 10$ for all data sets. The difference of the meta models' performance between $k' = 5$ and $k' = 10$ is very small in general, which is shown in Fig.5. Further, in the last column of Table I shows for each dataset the overall standard deviation of the secondary models for all k' values in the range $5 \leq k' \leq 10$. The small resulting values in all cases indicate the stability of the approach w.r.t. k' and validate the use of $k' = 5$ as the default value in our experiments.

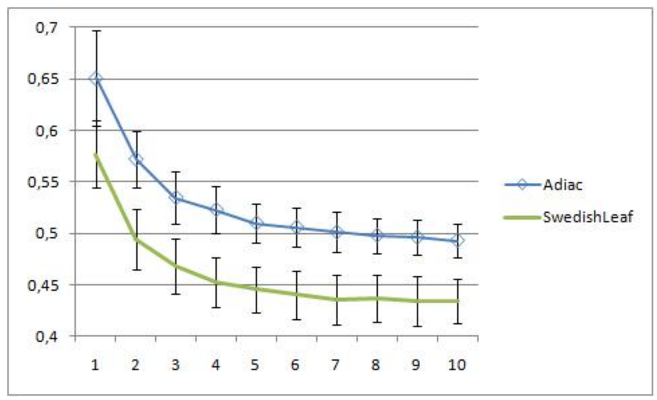

Figure 4. RMSE of the secondary models for various values of k' for Adiac and SwedishLeaf at $p = 5$ % noise. (Error bars show standard deviations.)

D. Execution Time

To investigate the overhead produced by IEP both in the off- and online computation, in Table III we report the execution times (on a Xeon 2.3 GHz processor) for three among the largest datasets: Wafer, Two-Patterns, and ChlorineConcentration. Please note that the off-line time refers to the time required for training the secondary regression level, which has to be performed only *once*. Online time refers to the actual time needed to classify a new time-series. IEP has almost the same off-line time (reported in minutes) as k-NN. This is because training is dominated by the classification of the hold-out set D_2 in both cases. Although the data sets in Table III are rather large, the resulting off-line time is reasonable in all cases. Regarding the online time, it is evident that IEP is able to maintain the fast classification of new time series.

[5]The time series whose labels were changed by this procedure, are assigned to an additional class (not included in the original data set). To keep our experimental evaluation meaningful, the time series with changed labels were excluded from the test set.

Dataset	size	p =1%			p =5%			p =10%			$\sigma_{k'}$
		IEP	1-NN	k-NN	IEP	1-NN	k-NN	IEP	1-NN	k-NN	
50 Words	905	**0.239** -/-	0.249	0.242	0.270	0.388	**0.260** -	**0.321** +/-	0.505	0.338	0.021
Adiac	781	**0.373** -/-	0.381	0.384	**0.415** +/-	0.508	0.451	**0.476** +/-	0.614	0.519	0.018
Car	120	0.279	**0.278** -	0.303	**0.310** +/-	0.416	0.353	**0.330** +/-	0.514	0.358	0.051
CBF	930	**0.004** +/+	0.106	0.047	**0.043** +/-	0.328	0.057	**0.139** +/+	0.496	0.174	0.044
ChlorineConcentration	4307	0.053	**0.021** +	**0.021** +	0.077	**0.075** -	**0.075** -	0.121	**0.115** -	**0.115** -	0.021
CinC	1420	**0.003** +/+	0.033	0.011	**0.008** +/-	0.143	0.021	**0.019** +/-	0.242	0.029	0.048
DiatomSizeReduction	322	**0.006** +/+	0.031	0.038	**0.010** +/+	0.141	0.049	**0.021** +/+	0.276	0.055	0.058
ECG200	200	**0.136** -/-	0.171	0.156	0.150	0.313	**0.134** -	0.201	0.442	**0.141** +	0.073
ECGFiveDays	884	**0.013** +/+	0.041	0.045	**0.020** +/+	0.164	0.136	**0.028** +/+	0.273	0.097	0.041
FaceFour	112	**0.063**	0.108	0.072	**0.075**	0.234	0.112	**0.099**	0.386	0.198	n/a
FacesUCR	2250	**0.029** +/+	0.059	0.039	**0.044** +/-	0.193	0.046	**0.070** +/-	0.316	0.083	0.028
Fish	350	**0.228** -/-	0.254	0.239	**0.244** +/-	0.386	0.280	**0.329** +/-	0.512	0.392	0.042
GunPoint	200	**0.010** -/+	0.036	0.061	**0.016** +/+	0.162	0.176	**0.043** +/+	0.258	0.207	0.107
Haptics	463	**0.490** +/-	0.582	0.532	**0.540** +/-	0.681	0.553	0.632	0.774	**0.622** -	0.026
InlineSkate	650	0.469	**0.461** -	0.483	**0.523** -/-	0.562	0.570	**0.602** +/+	0.679	0.661	0.025
ItalyPowerDemand	1096	**0.038** +/+	0.087	0.081	**0.059** +/-	0.237	0.060	**0.096** +/+	0.389	0.117	0.030
Lighting2	121	0.192	0.133	**0.125**	**0.209**	0.270	**0.209**	0.257	0.422	**0.239**	n/a
Lighting7	143	**0.254**	**0.254**	0.289	**0.279**	0.426	0.338	0.341	0.558	**0.310**	n/a
Mallat	2400	**0.014** +/-	0.055	0.018	**0.019** +/+	0.178	0.034	**0.048** +/-	0.308	0.067	0.030
MedicalImages	1141	**0.212** -/-	0.228	0.234	**0.228** +/-	0.339	0.256	**0.248** +/-	0.457	0.277	0.023
Motes	1272	**0.059** +/+	0.090	0.078	**0.073** +/+	0.206	0.107	**0.093** +/+	0.316	0.148	0.038
OSULeaf	442	0.320	**0.287** -	0.292	0.363	0.402	**0.345** -	0.407	0.523	**0.383** -	0.028
Plane	210	**0.005** +/+	0.034	0.038	**0.020** +/+	0.148	0.114	**0.021** +/+	0.304	0.225	0.093
SonyAIBORobotS.	621	**0.026** +/+	0.073	0.068	**0.035** +/+	0.234	0.083	**0.083** +/+	0.365	0.143	0.047
SonyAIBORobotS.II	980	**0.034** +/+	0.063	0.067	**0.037** +/-	0.212	0.119	**0.071** +/+	0.362	0.166	0.031
StarLightCurves	9236	0.076	0.119	**0.073** -	**0.096** +/-	0.253	0.098	0.151	0.388	**0.149** -	0.003
Symbols	1020	**0.023** +/-	0.061	0.031	**0.029** +/-	0.196	0.036	**0.066** +/-	0.332	0.069	0.040
SyntheticControl	600	0.020	0.076	**0.017** -	**0.028** +/-	0.227	0.058	**0.068** +/+	0.355	0.096	0.057
SwedishLeaf	1125	**0.170** +/+	0.206	0.197	**0.189** +/+	0.328	0.216	**0.247** +/-	0.461	0.257	0.023
Trace	200	**0.005** -/-	0.046	0.036	**0.005** +/-	0.180	0.064	**0.034** +/-	0.322	0.094	0.074
TwoLeadECG	1162	**0.001** +/+	0.041	0.052	**0.005** +/+	0.175	0.025	**0.025** +/-	0.315	0.028	0.042
TwoPatterns	5000	**0.001** +/+	0.065	0.007	**0.014** +/-	0.236	0.019	**0.068** +/+	0.384	0.079	0.026
Wafer	7164	**0.003** +/-	0.042	0.004	0.006	0.160	**0.005** +	**0.015** +/+	0.279	0.020	0.025
WordSynonyms	905	**0.224** -/-	0.238	0.241	**0.270** +/-	0.379	0.287	**0.332** +/-	0.510	0.349	0.032
Yoga	3300	**0.071** +/+	0.099	0.114	**0.085** +/-	0.223	0.115	**0.123** +/+	0.332	0.190	0.020

Table I
CLASSIFICATION ERROR.

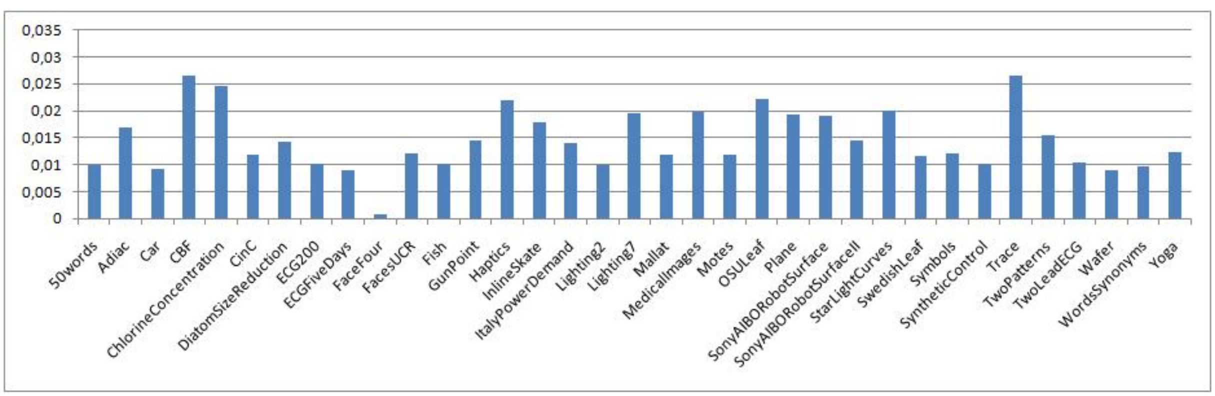

Figure 5. Average difference of the secondary models' performace (in RMSE) between using k'=5 and k'=10 for $p = 5$ % noise for each data set.

	Wafer	Two-Patterns	Chlorine
IEP	12.9 m / 0.22 s	19.8 m / 0.51 s	6.8 m / 0.23 s
k-NN	12.9 m / 0.06 s	19.8 m / 0.18 s	6.8 m / 0.04 s

Table III
EXECUTION TIMES: TRAINING TIME (OFFLINE) IN MINUTES AND PREDICTION TIME (ONLINE) IN SECONDS.

VI. CONCLUSION

We examined the problem of time-series classification based on the k-NN classifier and the DTW distance. Although the 1-NN classifier had been shown to be competitive, if not superior, to many state-of-the art time-series classification methods, we argued that in several cases we may not only consider $k > 1$ for the k-NN classifier, but also select k in an individual base for each time series that has to be classified.

We proposed an IEP mechanism that considers a range of k-NN classifiers (for different k values) and uses secondary regression models that predict the error of each such classifier. The proposed approach selects separately for each time series the classifier with the minimum predicting error. This allows for adapting to characteristics that are varying among the different regions in a data set and overcoming the problem of selecting a single k value.

Our experimental evaluation used a large collection of real data sets. Our results indicate that the proposed method is more robust and compares favorably against two examined baselines by resulting in significant reduction in the classification error. Other advantageous properties of the proposed method are its small sensitivity against the parameters it uses and the small overhead it adds in execution time.

It is important to state that the proposed IEP approach has several generic features. For the k-NN classifier, IEP can be employed for learning other parameters than k, such as the distance measure or the importance of nearest neighbors. More importantly, IEP is not only limited for the problem of k-NN classification of time-series data, since it can be used in combination with other classification algorithms and data types, whenever the complexity of the data requires such an individualized approach. Therefore, our future work involves the examination of IEP in a more general context of classification problems.

References

[1] K. Buza, L. Schmidt-Thieme, *Motif-based Classification of Time Series with Bayesian Networks and SVMs*, Proc. of 32nd Annual Conference of the Gesellschaft für Klassifikation (GfKl 2008), Springer Verlag, 2009.

[2] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, E. Keogh, *Querying and Mining of Time Series Data: Experimental Comparison of Representations and Distance Measures*, VLDB '08, 2008.

[3] C. Domeniconi and D. Gunopulos, *Adaptive Nearest Neighbor Classification using Support Vector Machines*, NIPS, 2001.

[4] C. Domeniconi, J. Peng. D. Gunopulos, *Locally Adaptive Metric Nearest-Neighbor Classification*, Transactions on Pattern Analysis and Machine Intelligence, 2002.

[5] N. Duffy, D. Helmbold, *Boosting Methods for Regression*, Machine Learning, 47, pp. 153–200, 2002.

[6] D. Eads, D. Hill, S. Davis, S. Perkins, J. Ma, R. Porter and J. Theiler, *Genetic algorithms and suppor vector machines for time series classification*, In Proc. of the International Society for Optical Engineering (SPIE), volume 4787, pages 74–85, 2002.

[7] P. Geurts *Pattern Extraction for Time Series Classification*, PKDD, LNAI 2168, pp. 115–127, Springer Verlag, 2001.

[8] T. Hastie and R. Tibshirani, *Discriminant Adaptive Nearest Neighbor Classification*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 18, No. 6, 1996.

[9] T. Hastie, R. Tibshirani, J. H. Friedman, *The Elements of Statistical Learning* Springer, 2009.

[10] A. K. Jain, R. C. Dubes, C.-C. Chen, *Bootstrap Techniques for Error Estimation*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-9, no. 5, 1987.

[11] E. Keogh, S. Kasetty, *On the Need for Time Series Data Mining Benchmarks: A Survey and Empirical Demonstration* SIGKDD, 2002.

[12] E.J. Keogh, M.J. Pazzani. Scaling up Dynamic Time Warping for Datamining Applications. *KDD* , pages 285 – 289, Boston, Massachusetts, USA, 2000.

[13] E. Keogh, C. Shelton, and F. Moerchen, *Workshop and Challenge on Time Series Classification*, Int. Conf. on Knowledge Discovery and Data Mining (KDD), 2007, http://www.cs.ucr.edu/˜eamonn/SIGKDD2007TimeSeries.html

[14] A. M. Molinaro, R. Simon, R. M. Pfeiffer, *Prediction error estimation: a comparison of resampling methods*, Bioinformatics, Vol. 21. no. 15, pp. 3301–3307, 2005.

[15] S. Ougiaroglou, A. Nanopoulos, A. N. Papadopoulos, Y. Manolopoulos, T. Welzer-Druzovec, *Adaptive k-Nearest-Neighbor Classification Using a Dynamic Number of Nearest Neighbors*, Advances in Databases and Information Systems, LNCS Vol. 4690/2007, Springer-Verlag, 2007.

[16] V. Petridis and A. Kehagias, *Predictive modular neural networks for time series classification*, Neural Networks, 10(1):31–49, 1997.

[17] M. Radovanovic and A. Nanopoulos and M. Ivanovic, *Time-Series Classification in Many Intrinsic Dimensions*, 10th SIAM International Conference on Data Mining

[18] C.A. Ratanamahatana, E. Keogh. Everything you Know about Dynamic Time Warping is Wrong. *SIGKDD W'shop on Mining Temporal and Seq. Data*, 2004.

[19] C.A. Ratanamahatana, E. Keogh. Making Time-series Classification More Accurate Using Learned Constraints. *SIAM Int'l Conf. on Data Mining*, 2004.

[20] T.M. Rath, R. Manmatha. Word image matching using dynamic time wrapping. *CVPR*, II:521–527, 2003.

[21] H. Sakoe, S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans. Acoustics, Speech, and Signal Proc.*, ASSP-26:43–49, 1978.

[22] P. Sykacek and S. Roberts *Bayesian time series classification*, Advances in Neural Processing Systems 14

[23] K. Tsuda, G. Rätsch, S. Mika, K.-R. Müller, *Learning to Predict the Leave-one-out Error of Kernel Based Classifiers*, Artificial Neural Networks, ICANN 2001, LNCS Vol. 2130/2001, Springer Verlag, 2001.

[24] D. Wettschereck, T. Dietterich, *Locally Adaptive Nearest Neighbor Algorithms*, Advances in Neural Information Processing Systems 6, Morgan Kaufmann Publishers, San Mateo, 1994.