A Localised Learning Approach Applied to Human Activity Recognition

Ahmed Y.A. Amer, Jean-Marie Aerts, Bart Vanrumste and Stijn Luca

***Abstract*—The recognition of human physical activities and postures based on sensor data has received much research attention in several human health and biomedical engineering applications. In this study, the challenges of class-imbalance and ambiguity (or confusion) are discussed that frequently arise in data from human activity recognition (HAR) systems. In order to reduce the influence of imbalance and ambiguity in HAR problems, a novel hybrid localised learning approach of K-nearest neighbours least-squares support vector machine (KNN-LS-SVM) is proposed. The classifier is applied to different synthetic and real-world datasets where imbalance and ambiguity are present. In this study, it is novel to apply a hybrid localised learning algorithm to the HAR problem. When compared to different global and local approaches, higher classification performances could be obtained by using the proposed localised learning approach. Furthermore, the computational effort could be reduced in an online learning mode.**

***Keywords*—local learning, class-imbalance, ambiguity, K-nearest neighbours algorithm, least-squares support vector machine.**

I. Introduction

Recognising human physical activities automatically via soft computing techniques is at the core of human activity recognition (HAR) studies. Human physical activities can be recognised by using computer vision techniques through the analysis of images and videos or by exploring sensory data that are obtained by wearable or portable sensors [1]. The importance of HAR systems is illustrated by the various amount of applications where they are used, e.g. medical monitoring [2], healthcare [3], military training, and sports [4]. Moreover, the use of HAR algorithms is enhanced by rapid advancements in sensor technology that enable to monitor people in their daily environments, e.g. by using smart-phones [5] or wearables [6]. In this study, we will focus on the problem of activity classification based on accelerometer data acquired through an inertial sensor-based HAR system.

For most of the real-world problems (e.g., HAR), data is not evenly distributed in the input space [7] which is a challenge to global learning algorithms in general (e.g., SVM, conventional and deep neural networks). In particular, we will discuss the problems of *class-imbalance* and *ambiguity* that frequently arise in data obtained from HAR systems and how they can influence the performance of a classifier. Class-imbalance occurs when instances from some activities are outnumbered by others. In case of extreme imbalance, the problem of rare events occurs (e.g. the detection of falls among several daily activities [8]). Ambiguity or confusion occurs when an activity is not clearly distinguishable from another one (e.g. eating might be confused with brushing teeth due to arm motion [9]). From input space perspective, ambiguity (confusion) could be a result of the overlap between the different classes or due to a highly nonlinear decision boundary between two closely spaced classes.

In order to reduce the influence of these problems (i.e. class-imbalance and ambiguity) on HAR systems, we introduce a localised learning approach. This approach is leading to a novel hybrid algorithm which is obtained from integrating the K-nearest neighbours (KNN) algorithm into a least-squares support vector machine (LS-SVM) algorithm, namely KNN-LS-SVM. In this approach, a classification model is built for each test example using only the training examples located in the vicinity of the test example. This novel localised approach of an LS-SVM algorithm is then applied to HAR problems. Which is the first time to apply a hybrid localised learning algorithm to the HAR problem.

In addition to handling the problems of class imbalance and ambiguity, we will show that the proposed kNN-LS-SVM has other advantages as well. These advantages include simplicity of implementation that can lead to a computational advantage compared to other classifiers (e.g., deep learning neural networks). Moreover it can deal with non-linearity due to the use of the LS-SVM method as opposed to a standard kNN. Our results indicate that the KNN-LS-SVM can be a suitable approach for the HAR application especially when applied to online problems and streaming analytics for which the data size is continuously increasing.

This article is structured as follows. In Section 2, an overview of related work on classification techniques for HAR systems and local learning is given. Section 3, gives a gentle introduction to LS-SVM and KNN-LS-SVM. Performances of LS-SVM and KNN-LS-SVM are compared in Section 4 using synthetic datasets. In Section 5, KNN-LS-SVM, as well as various global and local classifiers, are applied to four real-world datasets and their performances are compared to those of a benchmark study of a deep-learning approach. The obtained results are discussed in section 6. Finally, a conclusion is presented in Sections 7.

A.Y.A Amer and B. Vanrumste are with the KU Leuven, E-MEDIA, Department of Electrical Engineering (ESAT) STADIUS, (ESAT) TC, Campus Group T, Andreas Vesaliusstraat 13, 3000 Leuven, Belgium. E-mail: ahmed.youssefaliamer@kuleuven.be and bart.vanrumste@kuleuven.be

A.Y.A Amer and J.-M. Aerts are with the KU Leuven, Measure, Model & Manage Bioresponses (M3-BIORES), Department of Biosystems, Kasteelpark Arenberg 30 - box 2472 3001 Leuven, Belgium. E-mail: jean-marie.aerts@kuleuven.be

S. Luca is with Ghent University, Department of Data Analysis and Mathematical Modelling, Coupure links 653, 9000 Gent, Belgium. E-mail: stijn.luca@ugent.be

Manuscript received xxx, 2018; revised xxx, 2018.

II. Related Work

In this section, the state-of-the-art of the HAR problem and localised learning algorithms is introduced. Moreover, the benchmark study is briefly introduced.

A. State-of-the-art

A variety of classification algorithms have been applied to the problem of HAR, such as decision trees [10], [11], Naïve Bayes [10], Bayesian Networks [4], KNN [10], [11], convolutional neural networks [12], support vector machines (SVMs) [13], [3], and hidden Markov models (HMMs) [14]. Furthermore, deep learning approaches have recently gained much research attention [15], [16] and have been applied for HAR using low-power wearable devices [17], [18]. Moreover, transfer learning applied to deep neural networks for HAR application has recently received some attention in order to transfer models between different subjects. This approach is presented by *Renjie et al.* in their study [19] by applying the maximum mean discrepancy (MMD) algorithm to a two-layer convolutional neural network. However uncontrolled environment and online application of such an approach is still a challenge. Developing HAR models to be compatible with wearable systems is an important approach that is introduced by Cheng et al. in their work [20] namely InnoHAR model. This model is developed by concatenating convolution kernels of different scales and splicing with max-pooling layers. An important challenge that they are willing to tackle in their future work is class-imbalance in real-life human activities. For the purpose of real-time online data stream processing of HAR, the recent study by Amin et al. [21] developed a HAR model based on visual sensory data. The developed model is an optimised convolutional neural network (CNN) based model in which, deep features are extracted via a pre-trained CNN. The extracted features are fed to a deep Autoencoder (DAE) to learn the temporal behaviour of the signal and finally, the classification is done via a quadratic SVM.

From the literature review, there are some challenges that need to be handled. These challenges are class-imbalance [20], and applicability to real-world online modelling [19]. These challenges are motivating us to introduce a machine learning approach that can provide a high error performance regardless of the balance degree between the available instances of different activities. In addition, a low computational and temporal cost are desired in order to be suitable for its application in an online mode. Ultimately, we aim at introducing an approach that is compatible with streaming data analysis in which the modelling complexity is not affected by the continuously increasing size of the dataset.

Localised learning algorithms have been limited studied for HAR problems. A general framework for local learning was introduced in [22], where it was demonstrated that a localised approach might be very efficient to deal with the problems of imbalance and ambiguity. Among the most common local learning approaches are KNN algorithms which have been studied intensively in the context of HAR problems, e.g. [6], [23]. Zhang et al. [24] introduced a KNN-SVM algorithm that combined a KNN with an SVM for a visual object recognition problem. In [25] a more integrated framework, called localised support vector machine (LSVM), was introduced, which incorporates the neighbourhood information directly into SVM learning. The use of such localised approaches of SVMs, however, has not yet been studied in the context of HAR problems, nor its influence on the classification performance when imbalance or ambiguity is present.

The design of the localised hybrid algorithm KNN-LS-SVM aims to reduce the influence of imbalance and ambiguity in HAR problems. The choice of an LS-SVM classifier to be localised rather than a standard SVM is inspired by its computational advantage of solving a set of linear equations instead of solving the quadratic programming problem of standard SVM [26]. Moreover, an LS-SVM is considered as a very efficient global machine learning technique in many fields [27]. Based on synthetic and real-world data, the difference in performance between a KNN-LS-SVM and its alternative LS-SVM is illustrated for different degrees of class-imbalance and ambiguity.

B. Benchmark Study

In their study [18], *Ravi et al.* developed their model based on extracting shallow features in addition to deep learnt features via a CNN for HAR. The raw data from 3-axes accelerometer are segmented into time-windows with specific widths. From the extracted segments, deep learnt features and shallow features are extracted in parallel. Deep learnt features are extracted from the spectrogram of the segmented signal via the temporal convolutional layer. All extracted features, deep learnt and shallow features, are combined and fed to a fully connected layer whose output is propagated to the soft-max layer to be classified. The proposed approach in [18] outperformed other deep learning and ensemble approaches (MLP, J48 and logistic regression) [17], [28], [29] when applied to a set of published datasets [30], [31], [32], [33]. Because of its high performance and recency, the approach of [18] will be the benchmark of our study.

III. Local learning of SVMs

In this section, we start by reviewing the main concepts behind SVMs and localised approaches for SVMs. We will proceed by introducing our hybrid KNN-LS-SVM algorithm.

A. Support vector machines

SVMs are originally presented as binary classifiers, that assign each data instance $\mathbf{x} \in \mathbb{R}^d$ to one of two classes described by a class label $y \in \{-1, 1\}$ based on the decision boundary that maximises the margin $2/||w||_2$ between the two classes as shown in Figure 1 [34]. Generally, a feature map $\boldsymbol{\phi} : \mathbb{R}^d \mapsto \mathbb{R}^p$ is used to transform the geometric boundary between the two classes to a linear boundary $L : \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}) + b = 0$ in feature space, for some weight vector $\mathbf{w} \in \mathbb{R}^{p \times 1}$ and $b \in \mathbb{R}$. The class of each instance can then be found by $y = \operatorname{sgn}(w^\top \boldsymbol{\phi}(x) + b)$, where sgn refers to the sign function.

The estimation of the boundary L is performed based on a set of training examples $\mathbf{x}_i$ $(1 \leq i \leq N)$ with corresponding

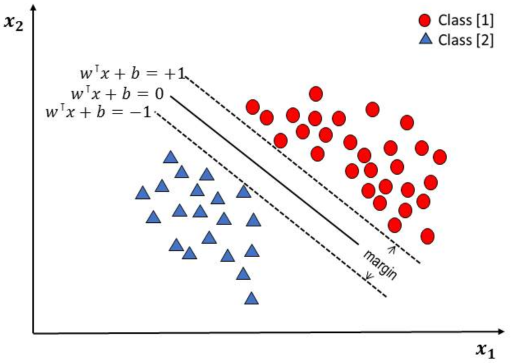


Fig. 1: Schematic representation of a two dimensional dataset consisting of two linearly separable classes. The dotted lines indicate the boundaries where the margin is maximised.

class labels $y_i \in \{-1, 1\}$. An optimal boundary is found by maximising the margin that is defined as the smallest distances between L and any of the training instances. In particular, one is interested in constants $\mathbf{w}$ and b that minimise a *loss-function*:

$$\min_{\mathbf{w},\, b;\, \xi} \frac{1}{2}\mathbf{w}^\top\mathbf{w} + C\sum_{i=1}^{N}\xi_i, \tag{1}$$

and are subject to:

$$y_i(\mathbf{w}^\top\boldsymbol{\phi}(\mathbf{x}_i) + b) \geq 1 - \xi_i \quad \text{and} \quad \xi_i \geq 0, \quad i = 1, 2, ..., N.$$

The constant C in (1) denotes the *penalty term* that is used to penalise missclassification through the slack variables ξ_i in the opimisation process.

The so-called *kernel-trick* avoids the explicit introduction of a feature map $\boldsymbol{\phi}$ and implicitly allows to use feature spaces of infinite dimensionality. A commonly used kernel is given by the Gaussian kernel:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\sigma_0^2}\right),$$

where σ_0 denotes the *kernel bandwidth*. Both σ_0 and C can be optimised as hyper-parameters in a cross-validation experiment.

LS-SVMs are obtained by using a least-squares error loss function [26]:

$$\min_{\mathbf{w},\, b;\, e} \frac{1}{2}\mathbf{w}^\top\mathbf{w} + \frac{1}{2}\gamma\sum_{i=1}^{N}{e_i}^2, \tag{2}$$

such that

$$y_i(\mathbf{w}^\top\boldsymbol{\phi}(\mathbf{x_i}) + b) = 1 - e_i, \qquad i = 1, 2, ..., N.$$

This optimisation procedure introduces errors e_i such that $1 - e_i$ is proportional to the signed distance of $\mathbf{x}_i$ from the decision boundary. In fact, the non-negative slack variable constraint is removed and the solution of the optimisation problem can be obtained by a set of linear equations, reducing computational effort [26].

B. localised LS-SVMs and KNN-LS-SVMs

In many HAR problems data are not evenly distributed in the input space. The presence of underrepresented data and severe class distribution skews affects the performance of learning algorithms that underly the HAR system [35]. Furthermore, the quality of a classifier further decreases when patterns are ambiguous, i.e., when they are not clearly belonging to one class or the other (i.e., ambiguity). Local learning approaches try to overcome such problems by building models that fit the data in the local neighbourhood around a test example and by locally adjusting to the properties of the data [22].

A well-known example of a local learning method is given by the KNN algorithm [36]. While nearest neighbours classifiers are very natural local learning methods, they suffer from the problem of high variance in the case of limited sampling. The use of a localised SVM can overcome such disadvantage as they often perform better than other classification methods in the neighbourhood consisting of a small number of examples ($k << N$) [24].

Furthermore, the complexity of global SVMs rapidly grows as the size of training instances increases. Besides, determining the right hyperparameters (kernel width and penalty term) of these models in a cross-validation experiment is computationally expensive. Local SVMs attempt to overcome these disadvantages by building small SVM models based on data in the local neighbourhood around a test example. This computational advantage is of particular importance in an online learning mode where one is interested to cheaply update the HAR model with the additional knowledge of a new data point. When using a global model, the model has to be recomputed from scratch, while for a local model only the training instances in the vicinity of the test examples matter.

While global SVMs consider the same weight for all training instances in the optimisation process (2), local learning approaches allow that the training samples near a test point are more influential than others. localised approaches of SVMs [25], [37] are based on weighting functions $\lambda(\mathbf{x}_s, \mathbf{x}_i)$ that express the similarity between the features vectors of the i-th data point $\mathbf{x}_i$ and a test instance $\mathbf{x}_s$. For an LS-SVM, this leads to the following cost function:

$$\min_{\mathbf{w},\, b;\, e} \frac{1}{2}\mathbf{w}^\top\mathbf{w} + \frac{1}{2}\gamma\sum_{i=1}^{N}\lambda(\mathbf{x}_s, \mathbf{x}_i)e_i^2, \tag{3}$$

such that

$$y_i(\mathbf{w}^\top\boldsymbol{\phi}(\mathbf{x}_i) + b) = 1 - e_i, \qquad i = 1, 2, ..., N.$$

Weighted least-squares support vector machines [38] use a similar approach, but here a different weighting function can be used for any given test point $\mathbf{x}_s$. In [37] the use of continuous similarity functions were studied including the Gaussian similarity criterion given by:

$$\lambda(\mathbf{x}_s, \mathbf{x}_i) = \exp\left(-\frac{||\boldsymbol{\phi}(\mathbf{x}_s) - \boldsymbol{\phi}(\mathbf{x}_i)||_2^2}{h^2}\right),$$

where $||.||_2$ denotes the Euclidean norm and h denotes a bandwidth parameter to be tuned. In this work we will study a binary valued similarity criterion:

$$\lambda(\mathbf{x}_s, \mathbf{x}_i) = \begin{cases} 1 & \text{if } ||\boldsymbol{\phi}(\mathbf{x}_s) - \boldsymbol{\phi}(\mathbf{x}_i)||_2 \leq r_s \\ 0 & \text{otherwise,} \end{cases}$$

where r_s is the K-th smallest distance among $\{||\boldsymbol{\phi}(\mathbf{x}_s) - \boldsymbol{\phi}(\mathbf{x}_j)||; 1 \leq j \leq N\}$. This formulation leads to the hybrid KNN-LS-SVM method that we will apply on HAR problems. In particular a classification model is built for each test example using only the training examples located in the vicinity of the test example [39].

In contrast to the localised LS-SVM proposed in [37], a KNN-LS-SVM has the additional advantage of sparseness. Indeed, for an LS-SVM or the localised version that uses a continuous similarity function all input data is required to construct the separating hyperplane [38]. This can be seen by solving the optimisation problem (2). Using the method of the Lagrangian multipliers, we find:

$$\mathcal{L}(w, b, e; \alpha) = \frac{1}{2}\|w\|_2^2 + \frac{1}{2}\gamma \sum_{i=1}^{N} \lambda(\mathbf{x}_s, \mathbf{x}_i)e_i^2 - \\ \sum_{i=1}^{N} \alpha_i(y_i[\mathbf{w}^\top \phi(\mathbf{x_i}) + b] - 1 + e_i),$$

where α_i are the *Lagrangian* multipliers. The optimality conditions are found by setting the first order partial derivatives to zero:

$$\frac{\partial \mathcal{L}}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^{N} \alpha_i y_i \boldsymbol{\phi}(\mathbf{x}_i),$$

$$\frac{\partial \mathcal{L}}{\partial b} = 0 \rightarrow \sum_{i=1}^{N} \alpha_i y_i = 0,$$

$$\frac{\partial \mathcal{L}}{\partial e} = 0 \Rightarrow \alpha_i = \gamma \lambda(\mathbf{x}_s, \mathbf{x}_i)e_i,$$

$$\frac{\partial \mathcal{L}}{\partial \alpha} = 0 \Rightarrow y_i(w^\top \boldsymbol{\phi}(\mathbf{x}_i) - b) = 1 - e_i, \forall 1 \leq i \leq N.$$

From the third condition, it is clear that the support values α_i are weighted by the similarity function and are zero when $\lambda(\mathbf{x}_s, \mathbf{x}_i) = 0$. Thus, for a KNN-LS-SVM the sparseness characteristic is returned to the LS-SVM. In an online learning mode, this sparseness will result in a computational advantage compared to LS-SVM, as we will show in Section V.

As shown in Figure 2, the algorithm of KNN-LS-SVM is implemented as follows:

1) Given a test example $\mathbf{x}_s$, compute distances to all training examples and pick the nearest K neighbours;
2) If all K neighbours would have the same label, assign the same label to $\mathbf{x}_s$.
3) Else, train the LS-SVM model with the K nearest neighbours.
4) Use the resulting classifier to label $\mathbf{x}_s$.

The parameter K and the distance metric (e.g. Euclidean, Mahalanobis or Chebyshev) are additional hyperparameters next to the kernel width σ_0 and the penalty term γ that are optimized in a cross validation approach.

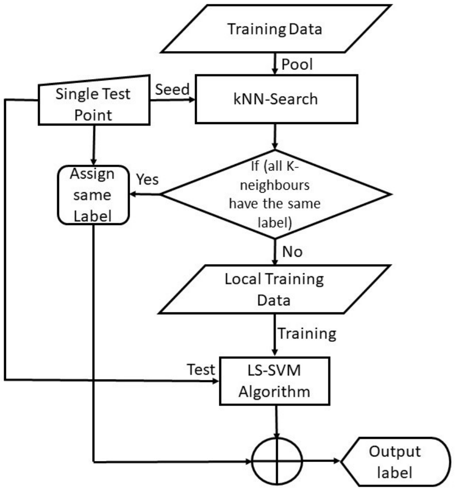


Fig. 2: A flow chart illustrating the localised learning algorithm of KNN-LS-SVM.

IV. Simulation experiments

The objective in this section is to present the problems of class-imbalance and ambiguity, that frequently arise in HAR problems, with controlled synthetic datasets. The generated synthetic datasets are two dimensional in order to simplify and visualise the problems which is not possible with high dimensional real-world datasets.

In this section, the performance of our KNN-LS-SVM method is compared with that of a global LS-SVM using three synthetic datasets. Two challenges are presented where global classifiers perform suboptimally: class-imbalance and ambiguity (or confusion). This is the first time to illustrate the problems of class-imbalance and ambiguity with synthetic data in the context of localised learning. Results are presented using the F_1 score that is defined as the harmonic average of precision and recall:

$$F_1 = 2\frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}.$$

In a two-class setting, recall and precision are defined as follows:

1) Recall is the ratio of instances that are correctly classified as positive to all positive instances;
2) Precision is the ratio of the instances that are correctly classified as positive to all instances classified as positive.

In case of multi-class setting, recall and precision are calculated based on one-vs-all approach (one class is positive and all other classes are negative).

The training of a local learning model requires the training of a local model for each individual test point. The selection of the hyperparameters of these local models is based on a cross-validation experiment where the accuracy (i.e. the ratio of correctly classified instances) is maximised. Accuracy-based model selection can handle the different distributions of the classes that can be present in a local region [32], [31]. In contrast with the global learning algorithms where the F_1-score is recommended for hyperparameters selection in case of class-imbalance, accuracy provides a balanced performance to the local learning algorithms as the majority/minority ratios changes in the local scale.

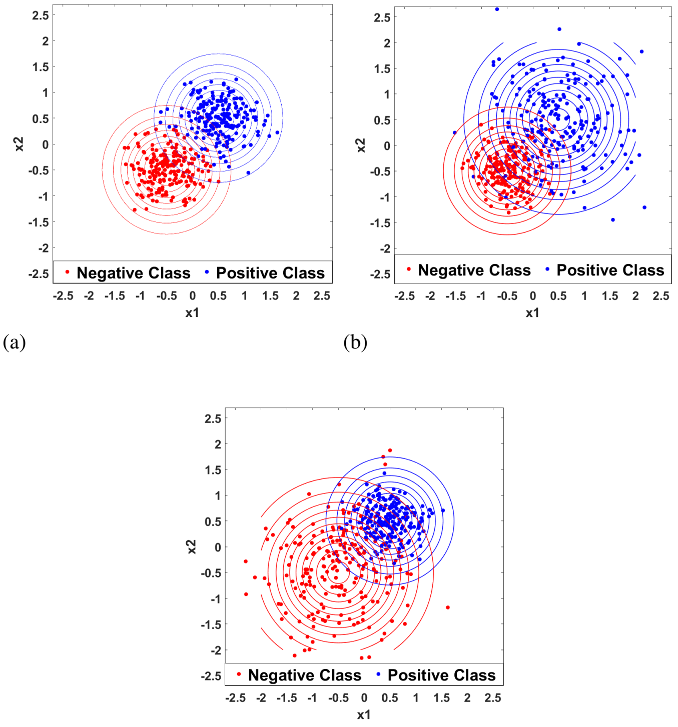
A. *Class-imbalance*

Learning from imbalanced data is still a focus of intense research, treating the problem of skewed class-distributions [40], [41]. It occurs when representatives of some classes appear much more frequently which poses a difficulty for learning algorithms, as they will be biased towards the majority group. In this section we study the use of a local learning method to deal with such imbalance.

A synthetic dataset is constructed consisting of data generated from two planar Gaussian distributions X_+ and X_-, that respectively represent a positive and a negative class, see Figure 3a. The distributions are centred at respectively $\mathbf{m}_+ = (\frac{1}{2}, \frac{1}{2})$ and $\mathbf{m}_- = (-\frac{1}{2}, -\frac{1}{2})$, with identical isotropic covariance matrices $\mathbf{\Sigma}_+ = \mathbf{\Sigma}_- = 0.35 I_2$, where I_2 denotes the identity matrix in $\mathbb{R}^{2\times 2}$. Experiments were performed where a number of $N = 400$ instances were simulated and the percentage p_+ of instances in the positive class X_+ varied in the range $p_+ \in \{50\%, 25\%, 12.5\%, 5, 2.5\%, 1.25\%\}$ with a number of instances $\{200, 100, 50, 20, 10, 5\}$ respectively. Both, an LS-SVM and a KNN-LS-SVM were trained with a Gaussian kernel. The models depend on hyperparameters (kernel width, penalty term and number of neighbours), the value of which are estimated in a 10-fold cross-validation experiment, where 80% of the simulated instances were used for training and 20% for testing. Figure 4a shows the F_1-scores averaged over the folds as a function of the imbalance percentage.

Clearly, the classifiers perform equally well when classes are balanced. However when there is class-imbalance the localised KNN-LS-SVM outperforms the global LS-SVM. Furthermore, the difference in performance tends to increase with an increasing degree of class-imbalance. At the percentages $50\%, 25\%, 12.5\%, 5\%, 2.5\%$, and 1.25%, the use of a KNN-LS-SVM results in a mean increasing difference in F_1-score of respectively $0.71\%, 1.17\%, 2.74\%, 7.74\%, 16.00\%$ and 19.68%. A statistical comparison of the F_1 scores using a paired t-test resulted in one-sided p-values: 0.4200, 0.4000, 0.2000, 0.0219, 0.0278, and 0.0276 respectively which show that the differences corresponding to an imbalance percentage at and above 5% were statistically significant with a significance level of 0.05.

The performance in case of class-imbalance can be further studied by changing the intra-class variance while fixing the inter-class variance leading to overlapping classes as shown in Figures 3b and 3c. Figures 4a, 4b, and 4c show respectively the difference in F_1-scores when $\mathbf{\Sigma}_+ = 2\mathbf{\Sigma}_- = 0.70 I_2$ (i.e., the variances of the positive class are as twice as large than the variances of the negative class) and when $\mathbf{\Sigma}_+ = \frac{1}{2}\mathbf{\Sigma}_- = 0.35 I_2$ (i.e., the variances of the positive class are as half as large than the variances of the negative class).



(a)

(b)

(c)

Fig. 3: Two-dimensional datasets consisting of two classes with data generated from Gaussian distributions $X_+ \sim N(\mathbf{m}_+, \mathbf{\Sigma}_+)$ and $X_- \sim N(\mathbf{m}_-, \mathbf{\Sigma}_-)$ with $\mathbf{m}_+ = (\frac{1}{2}, \frac{1}{2})$, $\mathbf{m}_- = (-\frac{1}{2}, -\frac{1}{2})$ and different covariance matrices: (a) $\mathbf{\Sigma}_+ = \mathbf{\Sigma}_- = 0.35 I_2$, (b) $\mathbf{\Sigma}_+ = 2\mathbf{\Sigma}_- = 0.70 I_2$ and (c) $\mathbf{\Sigma}_+ = \frac{1}{2}\mathbf{\Sigma}_- = 0.35 I_2$.

Clearly, the overall performance of both classifier decreases, when compared to the case where $\mathbf{\Sigma}_+ = \mathbf{\Sigma}_-$. The localised method, however, still outperforms the global LS-SVM for lower percentages p_+. In case, $\mathbf{\Sigma}_+ = 2\mathbf{\Sigma}_-$, there is a difference in F_1-score of $-1.17\%, 3.30\%, 8.82\%, 12.45\%, 51.59\%$, and 44.93% at the percentages $50\%, 25\%, 12.5\%, 5\%, 2.5\%$, and 1.25% respectively. A paired t-test showed that the differences were significant at the 0.05 level for imbalance percentages above 25% (with one-sided p-values 0.1310, 0.1300, 0.0236, 0.0296, 0.0204, and 0.0495 at the percentages $50\%, 25\%, 12.5\%, 5\%, 2.5\%$, and 1.25% respectively). In case, $\mathbf{\Sigma}_+ = \frac{1}{2}\mathbf{\Sigma}_-$, the performance curves of both classifiers remarkably decrease due to the overlap between the two classes. However, the KNN-LS-SVM still outperforms the LS-SVM with a difference in F_1-score of $0.48\%, 1.72\%, 34.92\%, 14.91\%, 26.85\%$, and 40.00% at the percentages $50\%, 25\%, 12.5\%, 5\%, 2.5\%$, and 1.25% respectively. The one-sided p-values of a paired t-test are 0.3500, 0.2445, 0.0034, 0.1498, 0.0160, and 0.0088 at the different imbalance percentages respectively. Thus, only at imbalance percentages 12.5%, 2.5% and 1.25% a significant difference at the 0.05 level was found. The significance is missed at the imbalance percentage of 5%, however the outperformance of the KNN-LS-SVM still present. Conclusively, the performance of the classifiers is not only influenced by the class-distribution imbalance but also by the intra-class variance and besides the inter-class variance in the input space.

B. Ambiguity

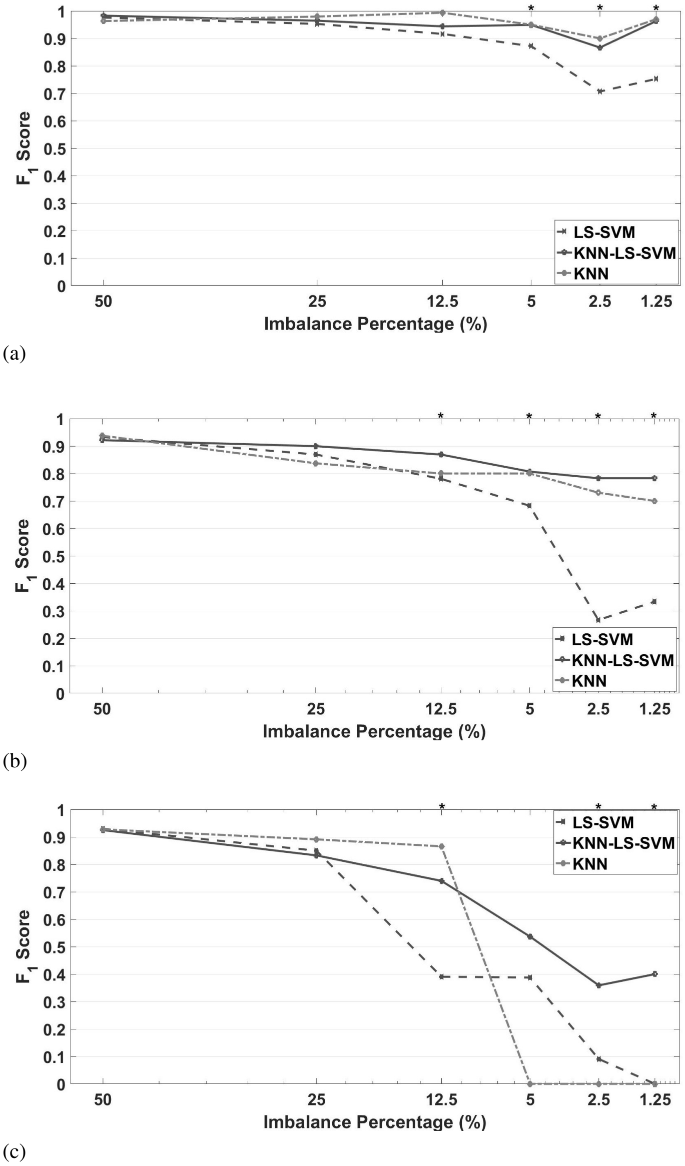
Ambiguity (or confusion) arises when regions exist in data space that are occupied by more than one class or when classes are very closely spaced [42]. In such cases, a global classifier will fit highly nonlinear boundaries that can become very complex on input space. A local learning algorithm attempts to locally adjust the complexity of the boundary to the properties of the data in each area of the input space.

In this section, we will study a simulated experiment where ambiguity occurs proportionally with the class-imbalance due to the discontinuity of the different classes' patterns and that is inspired from a general representation of Bottou & Vapnik [22], see Figure 5a. The data of 1000 instances is generated by mapping two variables $x1$ and $x2$ that are distributed according to a standard normal distribution $N(0, 1)$ to an univariate score $z = \sin(x1) * \sin(x2) + x1$. By setting specific ranges on the distribution of z different degrees of imbalance can be achieved. Where the percentage p_+ of instances in the positive class X_+ varied in the range $p_+ \in \{50\%, 33\%, 20\%, 10\%, 5\%, 2.5\%, 1.25\%\}$ with number of instances $\{500, 333, 200, 100, 50, 25, 12\}$ respectively. To simulate the set of p_+ of instances the range on z was chosen as:

$$0.5 < |z| < b, \text{ such that } P(0.5 < |z| < b) = p_+.$$

Hence, by varying b, the width of the positive class pattern and the number of the positive instances varies proportionally. Figure 5a shows an example of the boundary between the classes for $p_+ = 50\%$. The positive class is scattered in two stripes and surrounded by negative observations on both sides.

Figure 5b shows a comparison of the F_1-scores between an LS-SVM and a KNN-LS-SVM algorithm applied on the data set. For each percentage p_+, a 10-fold cross-validation experiment, similar as in Section IV-A, was performed. At the percentages 50%, 33%, 20% and 10%, the performance of both local and global classifiers is more or less the same. However, the use of a KNN-LS-SVM at the imbalance percentages of 5%, 2.5% and 1.25% lead to an increase in F_1-score of 10.86, 19.28 and 20.05% respectively. A paired t-test showed that these differences were all significant at a 0.05 levels (with one-sided p-values below 0.05) with one-sided p-values of 0.0245, 0.0150 and 0.0100 for $p_+ \in 5\%, 2.5\%, 1.25\%$ respectively. Clearly, the F_1-score rapidly decreases as the imbalance percentage increases. For higher imbalance percentages, the number of available instances from the positive class decreases making it hard for both classifiers to model the complex boundary between the classes. Ultimately, as shown in Figures 4 and 5, the KNN-LS-SVM can handle overlapping and nonlinearity problems together with class imbalance more efficiently than standard KNN.



(a)

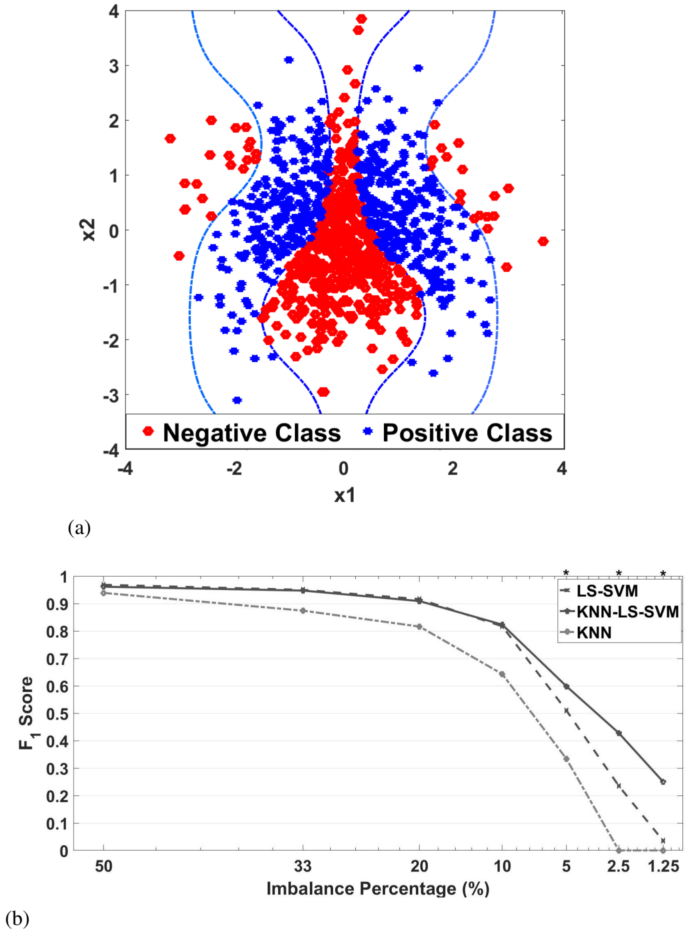
(b)

(c)

Fig. 4: (a) Averaged F$_1$-scores of a KNN-LS-SVM, KNN, and an LS-SVM obtained from a 10-fold cross-validation experiment using the synthetic dataset shown in Figure 3a and using different percentages p_+ of instances from the positive class.(b) F$_1$-scores of KNN-LS-SVM, KNN, and LS-SVM at different degrees of imbalance when $\mathbf{\Sigma}_+ = 2\mathbf{\Sigma}_- = 0.70I_2$. (c) F$_1$-scores of KNN-LS-SVM, KNN, and LS-SVM at different degrees of imbalance when $\mathbf{\Sigma}_+ = \frac{1}{2}\mathbf{\Sigma}_- = 0.35I_2$ and averaged over the runs of a 10-fold cross-validation experiment.The labels (*) indicated on top of the horizontal axis refer to the imbalance percentages where the difference in performance scores between KNN-LS-SVM and LS-SVM is statistically significant at the 0.05 level.

V. Real-world Data

As illustrated in section IV, class-imbalance is considered a challenge to global learning algorithms under some conditions



(a)

(b)

Fig. 5: (a) Two-dimensional non-linearly separable dataset of two classes where $p_+ = 50\%$. (b) F_1-scores of the KNN-LS-SVM, KNN, and LS-SVM averaged over the runs of a 10-fold cross-validation experiment. Performance scores are obtained by using the synthetic dataset describing confusion where the degree of imbalance is varied in the range $\{50\%, 33\%, 20\%, 10\%, 5\%, 2.5\%, 1.25\%\}$. The labels (*) indicated on top of the horizontal axis refer to the imbalance percentages where the difference in performance scores between KNN-LS-SVM and LS-SVM is statistically significant at the 0.05 level.

such as the imbalance ratio, the overlap between different classes, inter-class and intra-class variances, and ambiguity.

In this section, four real-world datasets are used to compare the performances of a KNN-LS-SVM to an LS-SVM [43], KNN, KNN-SVM [24], Profile SVM (PSVM) [25], Stacked Autoencoders (Stack-AE) [44]. Moreover, we compare our results with the results of a recent study of HAR that is based on the use of deep learning techniques [18]. Through the rest of the study, we will refer to the proposed algorithm of the benchmark study [18] as *Ravi (2017)*. Finally, the time performance of a KNN-LS-SVM is compared to those of the already mentioned classifiers except for *Ravi (2017)* due to the dedicated platform used in their study [18].

A. datasets

WISDM v1.1 The first dataset that we will study has been used to evaluate a system that uses phone-based accelerometers to perform HAR [32]. Several activities were recorded with different frequencies of occurrence: walking (38.6%), jogging (31.2%), walking upstairs (11.2%), walking downstairs (9.1%), sitting (5.5%), and standing (4.4%). Activities of 36 subjects were recorded using an impeded accelerometer of a smartphone with a sampling rate of 20 Hz and that was located in the front pocket. In this way, a total number of $1,048,576$ samples were acquired within approximately 14.56 recording hours. In a preprocessing phase, features were extracted as will be discussed later using a non-overlapping sliding window of 10 seconds.

Daphnet FoG This dataset contains annotated readings of 3 accelerometers attached to Parkinson's disease patients that experience freezing of gait (FoG) during walking tasks [31]. Since freezing of gait occurs rarely compared to other movement activities, the data is very imbalanced. Only $1/9$ of all recorded instances corresponded to the freezing-class. Sensors were attached to the shank (just above the ankle) and the thigh (just above the knee) using an elasticised strap and Velcro. A third sensor was attached to the lower back via a belt. The number of patients in this study is 10. The sampling rate of the accelerometers recordings was 64 Hz and the total number of the acquired samples is $1,917,887$ within approximately 8.32 recording hours. The features were extracted as will be discussed later from non-overlapping sliding windows of length 4 seconds.

WISDM v2.0 This dataset is used to evaluate a system that uses phone-based accelerometers to perform HAR [33]. Several activities were recorded with different frequencies of occurrence: walking (42.1%), jogging (14.7%), sitting (22.3%), standing (9.7%), Lying down (9.3%), and stairs (1.9%). Activities of 563 subjects were recorded using an impeded accelerometer of a smartphone with a sampling rate of 20 Hz and that was located in the front pocket. In this way, a total number of examples $2,980,765$ were acquired. In a preprocessing phase, features were extracted as will be discussed later using a non-overlapping sliding window of 10 seconds.

Skoda This dataset contains 10 manipulative gestures (classes) performed in a car maintenance scenario [30]. They are a subset of the 46 activities performed in the factory in one of the quality control checkpoints. Data is collected from one subject, with a sampling rate of 98 Hz. For comparison purpose, the accelerometer signals from one node are used (Node 16). The total number of samples is approximately $705,440$ samples. The ten classes will be shown in *Classification Performance* section.

B. Classification performance

We compare the classification performances of the KNN-LS-SVM to the LS-SVM, KNN, KNN-SVM, PSVM, Stacked Autoencoders (Stack-AE) and *Ravi (2017)*. For error performance evaluation, precision, recall and $F_1 - score$, are presented for the recognition of the activities present in the studied datasets. To make a consistent comparison with the recent study of HAR that is based on a deep learning approach, we use the same set of features of method *Ravi (2017)* [18] to train the KNN-LS-SVM, LS-SVM, KNN, KNN-SVM, PSVM, and Stack-AE. Several features are used: interquartile range, amplitude kurtosis, root mean square, variance, mean, standard deviation, skewness, minimum, median, maximum, mean-cross, and zero-cross.

Tables I, II,III, and IV show the recall/precision scores of the KNN-LS-SVM, LS-SVM, KNN, KNN-SVM, PSVM, Stack-AE and Ravi (2017) applied to the datasets WISDM v1.1, Daphnet FoG, WISDM v2.0 and Skoda respectively. Figures 6, 7, 8, and 9 show the F_1-*scores* of the classifiers applied to the real-world datasets.

TABLE I: Classification results of the KNN-LS-SVM, LS-SVM, *Ravi (2017)*, KNN, KNN-SVM, PSVM, and Stack-AE applied to WISDM v1.1.

		Walk	Jog	Sit	Stand	Walk Up	Walk Down
KNN-LS-SVM	Recall	99.36	96.70	97.97	99.04	93.88	95.34
	Precision	96.99	99.73	99.18	98.10	95.47	96.41
LS-SVM	Recall	99.51	99.87	96.25	96.19	94.22	90.39
	Precision	98.68	99.87	96.65	96.65	93.55	96.06
Ravi (2017)	Recall	99.37	99.40	98.56	97.25	95.13	95.90
	Precision	99.37	99.64	97.85	98.15	95.52	94.44
KNN	Recall	100	100	100	95.2	96.7	100
	Precision	98.9	100	96.77	100	100	100
KNN-SVM	Recall	82.7	89.12	86.40	95.23	80.36	74.00
	Precision	83.1	96.32	95.00	100	65.22	70.83
PSVM	Recall	78.05	96.15	96.30	100	55.56	56.41
	Precision	79.60	96.77	100	90.05	47.62	64.71
Stack-AE	Recall	98.50	98.70	90.30	89.50	87.30	74.30
	Precision	96.60	99.40	87.50	89.50	82.80	92.9

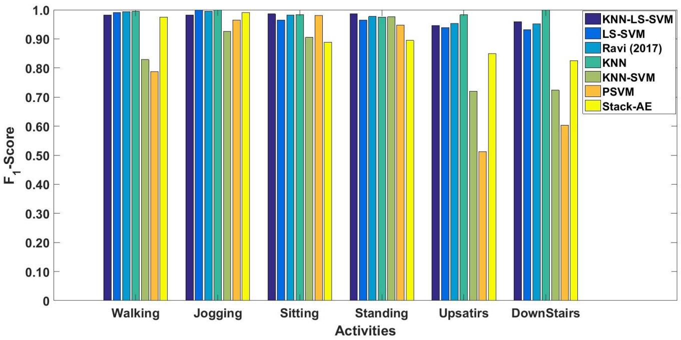


Fig. 6: F_1-*Scores* of the classifiers KNN-LS-SVM, LS-SVM, Ravi (2017), KNN, KNN-SVM, PSVM, and Stack-AE applied to WISDM v1.1

C. Time Performance

To compare the time performance between an LS-SVM and KNN-LS-SVM, we make use of a non-dedicated platform (i.e.,

TABLE II: Classification results of the KNN-LS-SVM, LS-SVM, Ravi (2017), KNN, KNN-SVM, PSVM, and Stack-AE applied to Daphnet FoG.

		Non-Freezing	Freezing
KNN-LS-SVM	Recall	97.79	72.92
	Precision	97.18	77.55
LS-SVM	Recall	98.31	62.66
	Precision	95.96	79.89
Ravi (2017)	Recall	98.15	59.92
	Precision	97.40	67.89
KNN	Recall	98.88	82.85
	Precision	98.34	82.32
KNN-SVM	Recall	97.80	60.00
	Precision	96.23	72.00
PSVM	Recall	91.94	54.72
	Precision	94.63	43.88
Stack-AE	Recall	97.00	70.12
	Precision	96.6	72.93

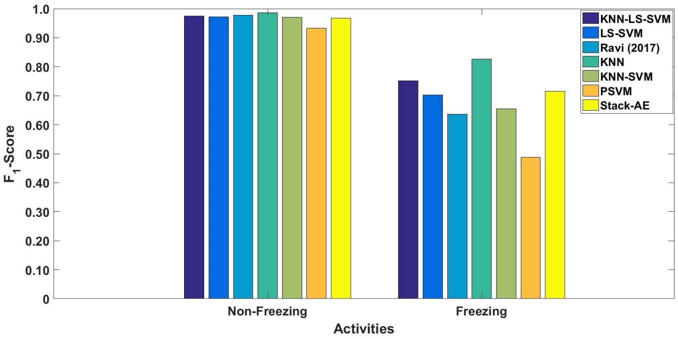


Fig. 7: F_1-*Scores* of the classifiers KNN-LS-SVM, LS-SVM, Ravi (2017), KNN, KNN-SVM, PSVM, and Stack-AE applied to Daphnet FoG

TABLE III: Classification results of the KNN-LS-SVM, LS-SVM, Ravi (2017), KNN, KNN-SVM, PSVM, and Stack-AE applied to WISDM v2.0.

		Walk	Jog	Sit	Stand	Lying Down	Stairs
KNN-LS-SVM	Recall	97.00	98.00	91.79	78.00	85.82	95.00
	Precision	96.60	97.41	86.62	92.00	88.80	97.96
LS-SVM	Recall	97.94	94.39	89.00	63.23	82.48	88.90
	Precision	95.96	98.54	80.59	92.45	83.09	88.90
Ravi (2017)	Recall	97.19	97.73	89.28	82.11	85.80	76.98
	Precision	97.17	98.01	87.32	82.05	88.65	85.00
KNN	Recall	96.53	92.83	82.46	66.20	42.45	61.76
	Precision	91.84	96.92	69.80	75.81	75.64	70.00
KNN-SVM	Recall	97.96	97.29	84.79	74.63	86.99	96.15
	Precision	96.64	98.17	88.81	79.37	76.98	96.17
PSVM	Recall	83.54	71.36	51.00	32.12	76.19	32.00
	Precision	90.88	56.72	72.11	70.97	34.78	15.69
Stack-AE	Recall	95.43	95.75	76.01	70.31	80.39	29.63
	Precision	92.79	96.21	79.22	73.77	70.29	100

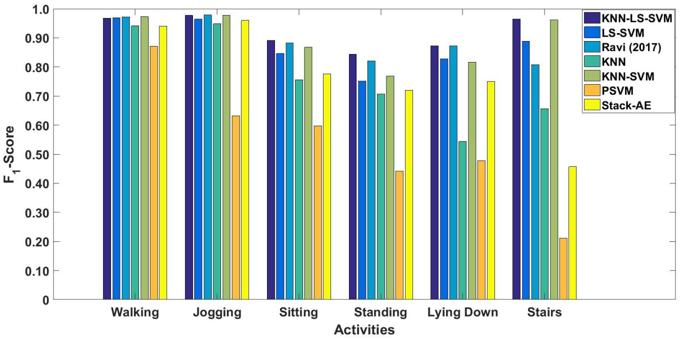


Fig. 8: F_1-*Scores* of the classifiers KNN-LS-SVM, LS-SVM, Ravi (2017), KNN, KNN-SVM, PSVM, and Stack-AE applied to WISDM v2.0

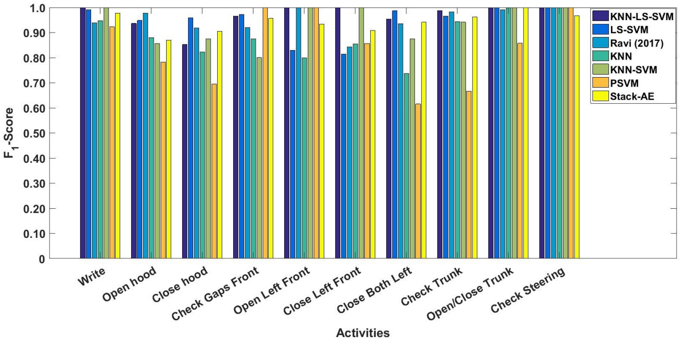


Fig. 9: F_1-*Scores* of the classifiers KNN-LS-SVM, LS-SVM, Ravi (2017), KNN, KNN-SVM, PSVM, and Stack-AE applied to Skoda dataset

System Type x64-based PC, Processor Intel(R) Core(TM) i7-6820HQ CPU @ 2.70GHz, 2701 MHz, 4 Core(s), 8 Logical Processor(s), Installed Physical Memory (RAM) 8,00 GB).

We compare the elapsed time that is required to run the algorithms in an online mode. In an online mode instances are evaluated one by one. For global classifiers, we measure the test time of one instance as in practice one model is trained and applied for all test instances. However, for the localised models (i.e. KNN, KNN-SVM and KNN-LS-SVM), the measured time includes both training and testing time. For PSVM, local models are trained offline by the training data points of each profile that result from partitioning the training set. Hence, the measured time for PSVM is test time only. The elapsed test times for each classifier applied to each dataset are depicted in Table V.

VI. Discussion

Applying the various classifiers on WISDM v1.1, we obtain the results shown in Table I and Figure 6. Notice that class-imbalance is not dominant in this dataset, except for *Sitting* and *Standing* activities (5.5 and 4.4% respectively). The best

TABLE IV: Classification results of the KNN-LS-SVM, LS-SVM, Ravi (2017), KNN, KNN-SVM, PSVM, and Stack-AE applied to Skoda dataset.

		Write	Open hood	Close hood	Check gaps Front	Open Left Front
KNN-LS-SVM	Recall	100	93.62	88.90	100	100
	Precision	100	93.62	81.84	93.33	100
		Close Left Front	Close Both Left	Check Trunk Gaps	Open and Close Trunk	Check Steering
	Recall	100	91.30	97.50	100	100
	Precision	100	100	100	100	100
LS-SVM		Write	Open hood	Close hood	Check gaps Front	Open Left Front
	Recall	100	96.97	95.08	94.59	89.47
	Precision	98.33	92.75	96.67	100	77.27
		Close Left Front	Close Both Left	Check Trunk Gaps	Open and Close Trunk	Check Steering
	Recall	73.33	97.37	96.55	100	100
	Precision	91.67	100	96.55	100	100
Ravi (2017)		Write	Open hood	Close hood	Check gaps Front	Open Left Front
	Recall	91.34	97.78	94.44	92.79	100
	Precision	96.67	97.78	89.47	91.15	100
		Close Left Front	Close Both Left	Check Trunk Gaps	Open and Close Trunk	Check Steering
	Recall	80.00	94.20	97.59	98.04	100
	Precision	88.89	92.86	98.78	100	100
KNN		Write	Open hood	Close hood	Check gaps Front	Open Left Front
	Recall	90.00	91.67	82.35	87.50	80.00
	Precision	100	84.62	82.35	87.50	80.00
		Close Left Front	Close Both Left	Check Trunk Gaps	Open and Close Trunk	Check Steering
	Recall	85.1	77.80	94.44	100	100
	Precision	85.71	70.00	94.44	100	100
KNN-SVM		Write	Open hood	Close hood	Check gaps Front	Open Left Front
	Recall	100	100	87.5	66.67	100
	Precision	100	75.00	87.50	100	100
		Close Left Front	Close Both Left	Check Trunk Gaps	Open and Close Trunk	Check Steering
	Recall	100	77.78	100	100	100
	Precision	100	100	88.90	100	100
PSVM		Write	Open hood	Close hood	Check gaps Front	Open Left Front
	Recall	100	100	80.00	100	100
	Precision	85.71	64.29	61.54	100	100
		Close Left Front	Close Both Left	Check Trunk Gaps	Open and Close Trunk	Check Steering
	Recall	75.00	44.44	50.00	90.00	100
	Precision	100	100	100	81.82	100
Stack-AE		Write	Open hood	Close hood	Check gaps Front	Open Left Front
	Recall	100	86.96	88.89	91.67	100
	Precision	95.65	86.96	92.31	100	87.50
		Close Left Front	Close Both Left	Check Trunk Gaps	Open and Close Trunk	Check Steering
	Recall	83.33	94.12	96.30	100	100
	Precision	100	94.12	96.30	100	93.75

TABLE V: The time performance of the classifiers KNN-LS-SVM, LS-SVM, KNN, KNN-SVM, PSVM, and Stack-AE applied to WISDM v1.1, Daphnet FoG, WISDM v2.0 and Skoda Datasets. The depicted results represent the consumed time in seconds to classify a single test point.

	WISDM v1.1	Daphnet FoG	WISDM v2.0	Skoda
KNN-LS-SVM	**0.0126**	**0.0014**	**0.0089**	**0.0035**
LS-SVM	0.0321	0.0062	0.0887	0.2283
KNN	25.32	34.00	19.77	16.822
KNN-SVM	19.78	14.017	20.10	12.88
PSVM	0.0290	0.0032	0.0111	0.0036
Stack-AE	0.0166	0.0171	0.0205	0.0248

performance is that of the KNN classifier for all activities except for *Standing*. The main drawback of the KNN classifier is the temporal complexity to get the optimum hyperparameters (i.e. K-number and distance metric). To obtain such good results, as shown in Table V, it takes approximately 25 seconds for one test point which is 2500 times the required time by KNN-LS-SVM. Moreover, KNN-LS-SVM provides the best performance for the minority classes of Sitting and Standing. We can notice here that KNN-LS-SVM is providing a performance that compromises between the superior performance of KNN with an expensive temporal complexity and LS-SVM with an acceptable error and time performance. The algorithm of KNN-LS-SVM is relying on KNN, but both the number K and the distance metric are globally optimised independently of the test set. For this dataset, we can claim that the error performance of KNN-LS-SVM is comparable in case of balanced activities and better for unbalanced activities. Moreover, KNN is not applicable to the online application as 25 seconds to classify a single point is more than the window size of 10 seconds that is used in this dataset.

For Daphnet FoG dataset, the class-imbalance is the dominant characteristic with a class-imbalance ratio of 1 to 9. As shown in Figure 7 and both Table I and Table V, KNN provides the best error performance and worst time performance due to hyperparameter optimisation. The second best error performance is the one of KNN-LS-SVM with best time performance which is $4 * 10^{-4}$ times that of KNN. The average run-time of KNN to classify one test point is 34 seconds which is not applicable for online classification as the window size for this dataset is only 4 seconds.

For WISDM v2.0, KNN-LS-SVM approximately provides the best error and time performance over all activities especially the extremely minor activity of Stairs (1.9%). The only competitive classifier is the KNN-SVM. However, its temporal and computational complexity is much higher than that of the KNN-LS-SVM. The superiority of the KNN-SVM and KNN-LS-SVM which are both based on an RBF kernel can be due to the presence of strong nonlinearity and overlapping classes in this dataset.

By applying the various classifiers to the Skoda dataset with 10 classes, KNN-LS-SVM provides the best performance for 6 classes. This dataset does not suffer class-imbalance such that KNN-LS-SVM is competing with the other global and local classifiers in the error performance. However, it outperforms the other classifiers in the time performance.

Ultimately, from Table V, it is obvious that KNN-LS-SVM provides the best time performance over all real-world datasets compared to the other classifiers which are implemented on the same platform while at the same time providing robustness against class imbalance.

VII. Conclusion

In this paper, we discussed the problems of class-imbalance and ambiguity that frequently arise in data obtained from HAR systems. A novel hybrid localised learning approach of KNN-LS-SVM is proposed to tackle these problems. Moreover, for the first time, these problems are analysed with synthetic datasets in the context of localised learning with a detailed illustration of the KNN-LS-SVM algorithm. In contrast to the already existing literature on HAR that mainly focuses on the use of global learning methods, we applied for the first time a hybrid localised learning algorithm to the problem of HAR. Furthermore, we compared the performance of the KNN-LS-SVM with other global and local learning techniques and the benchmark study of [18].

A localised method has the advantage to locally adjust the complexity of the decision boundary to the properties of the data in each area of the input space. The choice of LS-SVM instead of a standard SVM to be localised was motivated by the relatively computational simplicity of the LS-SVM compared to SVM [43]. This choice was further supported by the increased time performance in an online mode such that the KNN-LS-SVM has much potential to be suitable for online and streaming analytics problems in which the data size is continuously increasing.

Experiments using the synthetic data showed that the local classifier (i.e. KNN-LS-SVM) can be more robust against class-imbalance and ambiguity compared to a global classifier (i.e. LS-SVM). This was also confirmed by our experiments on the real-world data sets where the highest difference in the performance was obtained when the class-distribution was highly skewed.

The proposed algorithm is applicable to HAR applications as human health monitoring, e.g. fall detection, independent living of elderly, freezing of gait detection for Parkinson's patients, among others [45], [46], [31]. Experiments with real-world data illustrated the potential of the use of the localized approach for online and streaming analytics problems in HAR applications especially when applied to middle-sized data sets.

Moreover, the proposed localised approach has application potential to medical diagnostic problems that can suffer from class-imbalance problems, e.g. abnormality detection via screening, cancerous cells detection and Hyperthyroid diagnosis [47], [48], [49].

Finally, the proposed localised algorithm outperformed the benchmark global models under the following data-based conditions:

- The data sets contained up to $15,000$ data points.
- The imbalance percentages p_+ where the KNN-LS-SVM outperformed the LS-SVM ranged from 1.25% to 12.5%.
- Comparable performances were found for percentages p_+ given by 25% and 50%.
- The Fisher discriminant ratio that describes the overlap between classes could take values up to 4.08.

The Fisher discriminant ratio is defined as:

$$f = \frac{(m_+ - m_-)^2}{(\sigma_+^2 + \sigma_-^2)},$$

where m_+, m_- are the means of the positive and negative classes respectively. And σ_+, σ_- are the standard deviations of the positive and negative classes respectively.

In future research, we plan to apply and validate the method further in the context of real-time activity tracking of hospitalised patients.

Acknowledgment

This research is funded by a European Union Grant through wearIT4health project. The wearIT4health project is being carried out within the context of the Interreg V-A Euregio Meuse-Rhine programme, with EUR 2,3 million coming from the European Regional Development Fund (ERDF). With the investment of EU funds in Interreg projects, the European Union directly invests in economic development, innovation, territorial development, social inclusion and education in the Euregio Meuse-Rhine.

References

[1] A. Bulling, U. Blanke, and B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors." *ACM Computing Surveys*, vol. 1, pp. 1–33, 2014.

[2] L. Clifton, D. Clifton, M. Pimentel, P. Watkinson, and L. Tarassenko, "Predictive monitoring of mobile patients by combining clinical observations with data from wearable sensors," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, pp. 722–730, 2014.

[3] S. Luca, L. Vuegen, H. Van hamme, P. Karsmakers, and B. Vanrumste, *Decision support systems for home monitoring applications: Classification of activities of daily living and epileptic seizures*. London: IET, Institution of Engineering and Technology, 2016, ch. 13, pp. 271–291.

[4] O. Lara and M. Labrador, "A mobile platform for real time human activity recognition," in *Proc. IEEE Conference on Consumer Communications and Networks*, 2012.

[5] J. Morales and D. Akopian, "Physical activity recognition by smartphones, a survey," *Biocybernetics and Biomedical Engineering*, vol. 37, no. 3, pp. 388–400, 2017.

[6] F. Attal, S. Mohammed, M. Dedabrishvili, F. Chamroukhi, L. Oukhellou, and Y. Amira, "Physical human activity recognition using wearable sensors," *Sensors*, vol. 15, no. 12, pp. 31 314—-31 338, 2015.

[7] V. Vapnik and Bottou, "Local algorithms for pattern recognition and dependencies estimation." *Neural Computation*, vol. 5, pp. 893–909, 1993.

[8] G. Baldewijns, G. Debard, G. Mertes, B. Vanrumste, and T. Croonenborghs, "Bridging the gap between real-life data and simulated data by providing a highly realistic fall dataset for evaluating camera-based fall detection algorithms," *Healthcare Technology Letters*, vol. 3, no. 1, pp. 6–11, 2016.

[9] O. Lara and M. A.L.., "A survey on human activity recognition using wearable sensors." *IEEE Communications Surveys and Tutorials*, vol. 15, pp. 1192–1209, 2013.

[10] L. Jatoba, U. Grosmann, C. Kunze, J. Ottenbacher, and W. Stork, "Context-aware mobile health monitoring: Evaluation of different pattern recognition methods for classification of physical activity," in *Proceedings of the 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2008, p. 5250–5253.

[11] U. Muarer, A. Smailagic, D. Siewiorek, and M. Deisher, "Activity recognition and monitoring using multiple sensors on different body positions," in *Proc. International Workshop on Wearable and Implantable Body Sensor Networks*, Washington, DC, USA, 2006.

[12] Z. Ming, L. T. Nguyen, Y. Bo, J. Ole, Z. Jiang, W. Pang, and Z. Joy, "Convolutional neural networks for human activity recognition using mobile sensors," in *Proceedings of the 6th International Conference on Mobile Computing, Applications and Services*, vol. 6, 2014.

[13] Z.-Y. He and L.-W. Jin, "Activity recognition from acceleration data using ar model representation and svm," in *International Conference on Machine Learning and Cybernetics*, 2008, p. 2245–2250.

[14] C. Zhu and W. Sheng, "Human daily activity recognition in robotassisted living using multi-sensor fusion," in *IEEE International Conference on Robotics and Automation*, 2009, p. 2154–2159.

[15] R. Miotto, F. Wang, S. Wang, X. Jiang, and J. T. Dudley, "Deep learning for healthcare: review, opportunities and challenges," *Briefings in Bioinformatics*, p. bbx044, 2017.

[16] D. Wu, N. Sharma, and M. Blumenstein, "Recent advances in video-based human action recognition using deep learning: A review," in *International Conference on Nueral Networks*, 2017, pp. 2865–2872.

[17] D. Ravi, C. Wong, B. Lo, and G.-Z. Yang, "Deep learning for human activity recognition: A resource efficient implementation on low-power devices," in *Proc. 2016 IEEE 13th Int. Conf. Wearable Implantable Body Sensor Network*, 2016, pp. 71–76.

[18] ——, "A deep learning approach to on-node sensor data analytics for mobile or wearable devices." *IEEE Journal of Biomedical and Health Informatics*, vol. 21, pp. 56–64, 2017.

[19] R. D. et al., "Empirical study and improvement on deep transfer learning for human activity recognition." *Sensors*, 2019.

[20] C. X. et al., "Innohar: a deep neural network for complex human activity recognition." pp. 9893–9902, 2019.

[21] U. A. et al., "Action recognition using optimized deep autoencoder and cnn for surveillance data streams of non-stationary environments." *Future Generation Computer Systems*, pp. 386–397, 2019.

[22] L. Bottou and V. Vapnik, "Local learning algorithms." *Neural computation*, vol. 4, pp. 888–900, 1992.

[23] P. Paul and T. George, "An effective approach for human activity recognition on smartphone," in *2015 IEEE International Conference on Engineering and Technology (ICETECH)*, 2015, pp. 1–3.

[24] H. Zhang, A. Berg, M. Maire, and J. Malik, "SVM - KNN: discriminative nearest neighbor for visual object recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2006, pp. 1–3.

[25] H. Cheng, P.-N. Tan, and R. Jin, "Localized support vector machine and its efficient algorithm," in *Proceedings of the 2007 SIAM International Conference on Data Mining*, 2007.

[26] J. Suykens and J. Vandewalle, "Least squares support vector machine classifiers." *Neural processing letters*, vol. 9, pp. 293–300, 1999.

[27] Y. Huang and H. Ma, "Application of least square support vector machine in electronic engineering based on principal component analysis," in *Advances in Mechanical and Electronic Engineering*, D. Jin and S. Lin, Eds. Berling, Heidelberg: Springer, 2012.

[28] M. A. Alsheikh, A. Selim, D. Niyato, L. Doyle, S. Lin, and H.-P. Tan, "Deep activity recognition models with triaxial accelerometers." in *AAAI Workshop: Artificial Intelligence Applied to Assistive Technologies and Smart Environments*, 2016.

[29] C. Catal, S. Tufekci, E. Pirmit, and G. Kocabag, "On the use of ensemble of classifiers for accelerometer-based activity recognition," *Applied Soft Computing*, vol. 37, pp. 1018–1022, 2015.

[30] Z. P. et al., "Activity recognition from on-body sensors by classifier fusion: sensor scalability and robustness." 2007.

[31] M. Bachlin, M. Plotnik, D. Roggen, I. Maidan, J. Hausdorff, N. Giladi, and G. Troster, "Wearable assistant for parkinson's disease patients with the freezing of gait symptom," *IEEE Transactions on Information Technology in Biomedicine*, vol. 14, no. 2, pp. 436–446, 2010.

[32] J. Kwapisz, G. Weiss, and S. Moore, "Activity recognition using cell phone accelerometers," in *Proceedings of the Fourth International Workshop on Knowledge Discovery from Sensor Data*, O. Omitaomu, V. Chandola, A. Ganguly, J. Gama, R. Vatsavai, M. Gaber, and N. Chawla, Eds., 2010, pp. 10–18.

[33] L. J. W. et al., "Design considerations for the wisdm smart phone-based sensor mining architecture." pp. 25–33, 2011.

[34] B. Schölkopf, J. Platt, J. Shawe-Taylor, A. Smola, and R. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Computation*, vol. 13, no. 7, pp. 1443–1471, 2001.

[35] H. Haibo and E. Garcia, "Learning from imbalanced data," *IEEE Transactions on knowledge and data engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[36] A. Webb and K. Copsey, *Statistical Pattern recognition*, 3rd ed. Wiley, UK, 2011.

[37] Z. Karevan, Y. Feng, and J. Suykens, “Moving least squares support vector machines for weather temperature prediction,” in *Proceedings of European Symposium on Artificial Neural Networks*, 2017.

[38] D. B. J. Suykens, J.A.K., L. Lukas, and J. Vandewalle, “Weighted least squares support vector machines: robustness and sparse approximation,” *Neurocomputing*, vol. 48, no. 1-4, pp. 85–105, 2002.

[39] A. Amer, “Localised least squares support vector machines with application to weather forecasting,” Master’s thesis, KU Leuven, 2016.

[40] H. He and Y. Ma, *Imbalanced Learning: Foundations, Algorithms, and Applications*, 1st ed. Wiley-IEEE Press, New York, 2013.

[41] B. Krawczyk, “Learning from imbalanced data: open challenges and future directions,” *Progress in Artificial Intelligence*, vol. 5, no. 4, pp. 221–232, Nov 2016.

[42] T. Trappenberg and A. Back, “A classification scheme for applications with ambiguous data,” in *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, vol. 6, 2000, pp. 296–301.

[43] J. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle, *Least Squares Support Vector Machines*. Singapore: World Scientific Publishing Co., 2002.

[44] C. D. et al., “A practical tutorial on autoencoders for nonlinear feature fusion taxonomy, models, software and guidelines,” *Information Fusion*, pp. 78–96, 2018.

[45] Y. Choi, A. Ralhan, and S. Ko, “A study on machine learning algorithms for fall detection and movement classification,” in *Information Science and Applications (ICISA), 2011 International Conference on*. IEEE, 2011, pp. 1–8.

[46] Q. Ni, A. B. García Hernando, and I. P. de la Cruz, “The elderly’s independent living in smart homes: A characterization of activities and sensing infrastructure survey to facilitate services development,” *Sensors*, vol. 15, no. 5, pp. 11 312–11 362, 2015.

[47] C. Tataru, D. Yi, A. Shenoyas, and A. Ma, “Deep learning for abnormality detection in chest x-ray images,” 2017.

[48] K. Kourou, T. P. Exarchos, K. P. Exarchos, M. V. Karamouzis, and D. I. Fotiadis, “Machine learning applications in cancer prognosis and prediction,” *Computational and structural biotechnology journal*, vol. 13, pp. 8–17, 2015.

[49] I. M. D. Maysanjaya, H. A. Nugroho, and N. A. Setiawan, “A comparison of classification methods on diagnosis of thyroid diseases,” in *Intelligent Technology and Its Applications (ISITIA), 2015 International Seminar on*. IEEE, 2015, pp. 89–92.