Siamese Neural Network based Similarity Metric for Inertial Gesture Classification and Rejection

Samuel Berlemont, Grégoire Lefebvre, Stefan Duffner, Christophe Garcia

Siamese Neural Network based Similarity Metric for Inertial Gesture Classification and Rejection

Samuel Berlemont[1,2], Grégoire Lefebvre[1], Stefan Duffner[2] and Christophe Garcia[2]
[1] Orange Labs, R&D, F-38240 Meylan, France
[2] Université de Lyon, CNRS INSA-Lyon, LIRIS, UMR5205, F-69621, France
[1]{*firstname.surname*}@*orange.com*, [2]{*firstname.surname*}@*liris.cnrs.fr*

***Abstract*— In this paper, we tackle the task of symbolic gesture recognition using inertial MicroElectroMechanicals Systems (MEMS) present in Smartphones. We propose to build a non-linear similarity metric based on a Siamese Neural Network (SNN), trained using a new error function that models the relations between pairs of similar and dissimilar samples in order to structure the network output space. Experiments performed on different datasets regrouping up to 22 individuals and 18 gesture classes, targeting the most likely real case applications, show that this structure allows for an improved classification and a higher rejection quality over the conventional MultiLayer Perceptron (MLP) and Dynamic Time Warping (DTW) similarity metric.**

I. INTRODUCTION

Nowadays, inertial sensors are present in most existing Smartphones and many other handheld devices. While the accelerometer keeps track of the linear accelerations of the device in the 3D space, the gyrometer measures the angular velocities. These synchronized signals are classically used in services such as portrait/landscape screen rotation or for gesture recognition. Three main applications can be then identified in order to trigger a predetermined functionality: posture recognition (*i.e. flipping, hanging the phone, etc.*) ; activity recognition (*i.e. walking, jogging, biking, etc.*) ; and dynamic gesture recognition (*i.e. when the user "draws" a symbolic gesture in the air, e.g. a circle, a square, etc.*). While posture recognition is relatively straightforward, models used in gesture and action recognition have to face multiple challenges. On the one hand, inertial MEMS present inherent flaws that have to be taken into account, since they can be deceived by physical phenomena. On the other hand, in a real open-world application, inertial based gesture recognition systems have to deal with high variations between users (*i.e. right/left-handed users, dynamic/slow movements, users in mobility, etc.*). To offer more functionality to final users, such a system should propose a large vocabulary of interaction and reject all decision uncertainties and parasite motions.

In this paper, we propose a novel gesture classification and rejection method based on Siamese Neural Networks (SNN). This method learns simultaneously auto-extracted features in order to be more robust to physical phenomena and a non-linear similarity metric for dealing with numerous gesture categories. Moreover, we investigate two separate kinds of rejections: the first one consists in rejecting samples from known classes whose classification is too uncertain, and the second type of rejection concerns samples from unknown classes, showing the ability of our model to process unknown samples in a sensible manner and isolate them.

This paper is organized as follows. Section II presents related works on gesture recognition and rejection criteria. In Section III, we quickly sum up the MLP theory and notations in order to introduce the SNN, with details of our modified backpropagation algorithm and training strategy. Section IV describes our experimental setups and results when comparing our solution to MLP and DTW based similarity metric. Finally, our conclusions are drawn and perspectives are presented.

II. RELATED WORK

Inertial gesture recognition has been researched for the past ten years, and three main strategies can be identified. The first strategy relies on geometric similarity metrics combined with a direct classifier which compares the sample to be recognized to a gallery of references. Its main representative [1] is a model constructed from the Dynamic Time Warping (DTW) similarity distance, suited for time-series, and a K-Nearest Neighbor (K-NN) classifier. The second strategy [11] consists in a statistical modeling approach, with the application of Hidden Markov Models (HMM). Finally, the last strategy implies the use of kernel-based models learned from features, such as Support Vector Machines (SVM) [14] or Bayesian Networks [3]. A more precise description for each approach can be found in [10]. In our study, we focused on the geometric similarity metrics and the neural-based strategies, as well as their rejection criteria. In [4], Choe *et al.* apply a DTW-based model to inertial gesture recognition using mobile phones. The authors test a KNN classifier based on templates generated from a dataset gathering 4 subjects and 20 gestures for a total of 2000 samples. The use of a limited number of templates implies a reduced computational cost for recognition at around 90% precision, similar to the case where each sample is used as a template. Moreover, thresholds depending on the average and standard deviation of intra-class distances are used, allowing for class-specific rejection. Neural network-based strategies for inertial gesture recognition are less frequent. In [10], Lefebvre *et al.* apply a bi-directional long short term memory (BLSTM) recurrent neural network on our dataset of 14 classes and 22 users. A 95.18% accuracy is reached for a multi-user configuration, while Duffner *et al.* [6] applied a convolutional neural

network to the same dataset, with correct recognition rates of 97.9% and 93.4% respectively for user-dependent and user-independent configurations, proving the relevance of neural networks for gesture classification.

The notion of rejection in classification has been studied in other areas and applications. Two kinds of rejection criteria have been proposed in the literature, with the first criterion based on the actual input signals of the network, and the second based on decision boundaries for the output space. Following the first strategy, Vasconcelos *et al.* [13], tackling handwritten digit recognition, suggest using "guard units" for each class. These units are defined by their weight vector, which is composed by the means of the features for every training pattern belonging to the class. Therefore, after the activation of the network by a new sample, the guard units check a similarity score between the input sample and each class, issuing a "0" output for neurons corresponding to the classes that do not meet the rejection criterion. For an input sample I and a weight vector W corresponding to the class of the sample, the scalar product $I.W$ should be closer to the norm of W than the inner product for a sample belonging to a different class. The rejection criterion is then defined by a threshold ρ where the input is accepted by a class i only if $I.W_i \geq (W_i.W_i - \rho)$.

The second strategy is a lot more represented, and can be subdivided into threshold-based and custom boundaries determination methods. Fels *et al.* [7] apply the MLP model to the "Data-Glove" to produce a hand-gesture-to-speech system. Based on the angles between the fingers as well as the position and orientation of the hand, 5 MLPs are trained and combined to represent a vocabulary of 203 words, constructed on 66 "root words". Respectively 8912 and 2178 samples were used for the training and testing phases. In order to preserve the natural aspect of the interaction, a special interest is devoted to limiting the number of errors. A thresholding strategy on the value of the highest Softmax output is adopted, for a final actual error rate of 0.96%, and a mean rejection rate of 2.25%. In [12], Singh *et al.* propose an additional step to improve this rejection method. Applied to object recognition using a sequence of still images from the Minerva benchmark, their rejection criterion relies on generated patterns. For each feature, given μ the mean and σ the standard deviation of the training samples, random numbers are drawn between $\mu - 2.5\sigma$ and $\mu + 2.5\sigma$, and removed if comprised between min and max. Thus, the generated patterns represent the outside boundaries of each class, and are trained to produce outputs close to zero for every class. Test samples are then classically rejected if all of their outputs are under a 0.5 threshold. A thresholding on the maximum output corresponds to a spheric reliability zone.

In order to define more flexible boudaries, Gasca *et al.* [8] propose to estimate hyperplanes emulating the decision boundaries in the MLP output space in order to identify "overlap" regions, where the samples are more likely to be misclassified. The MLP is combined with a K-NN classification, based on the outputs of the training samples correctly classified after training. When recognizing a pattern, from the two nearest classes, the label is accepted only if the class given by the network matches the one selected by the K-NN, given the sample is not in the overlap area between hyperplanes. Experiments carried out using the databases from the repository of University of California show a decrease error of 50% on the test set, which is explained by the need to select representative samples for the hyperplanes definition.

In the light of the state-of-the-art, we assessed the rejection potential of the SNN when building a non linear similarity metric between pairs of samples. Two classical models were chosen to compare the performance of this model. The DTW-based model stood out as the best immediate comparison with another similarity metric. Finally, we decided to compare the SNN to the MLP to evaluate their performances as neural networks.

III. Proposed model

Inspired by cognitive science, we propose a model based on SNN to recognize symbolic gestures on Smartphones. This non-linear learning strategy is crucial to classify our gesture vocabulary and to reject others gestures and false alarms. While the MLP is a classifier, the main idea behind the SNN is to build a non linear similarity metric from multiple samples. Thus, although the SNN still keeps the computational parts of the classical MLP, it essentially differs from it by an original training strategy with a new error function for backpropagation. An SNN learns to produce feature vectors from pairs of samples that are discriminative for the final classification.

A. MultiLayer Perceptron

An MLP is a feed-forward network composed of multiple computational neural layers whose behavior mirrors our understanding of brain neurons: the input layer is directly activated by the gesture sample to be recognized, with one artificial neuron for each dimension ; the hidden layers hold the computational power of the network; the output layer is formed of one neuron for each training class. The output neuron with the strongest activation determines the winning class for that sample.

Let x_i be the i^{th} dimension of the input sample x, a_i^L the activation of the i^{th} neuron n_i of the layer L, I_i^L the input of the i^{th} neuron of the layer L, φ the activation function for each neuron, and ω_{ji} the connection weight between the neurons from two consecutive layers, of respective indexes j and i. For the input layer, the activations of the neuron n_i is equal to the i^{th} feature of the input sample. For any other layer L, we then define :

$$\begin{cases} I_i^L = \sum_{j=1}^{n} \omega_{ji} a_j^{L-1} \\ a_i^L = \varphi(I_i^L) \end{cases} \quad (1)$$

The MLP training is performed using the backpropagation algorithm. Following a gradient descent logic, for each training sample, the network is activated, then the discrepancy

between the activations of the output layer neurons and the target output is computed. The main error criterion is based on the cross-entropy between the estimate and the target distributions for the model. Let $\mathbf{X} = \{x^1, ..., x^N\}$ the set of training samples, K the number of classes, t_{kn} the target for the neuron k of the sample x^n, and y_{kn} the corresponding network output, then the error E_W, with $\mathbf{W} = \{\omega_{ji}\}$ is defined as follows:

$$E_W = -\sum_{n=1}^{N}\sum_{k=1}^{K} t_{kn} log(y_{kn}). \quad (2)$$

Moreover, given the learning rate λ, the set of weights $\mathbf{W^t}$ at the epoch t is then updated following Equation (3).

$$\omega_{ji}^{t+1} = \omega_{ji}^{t} - \lambda \frac{\partial E}{\partial \omega_{ji}}(\omega_{ji}^{t}). \quad (3)$$

This error is propagated backwards in the network in order to update each connection weight, following the *delta* rule, with $\delta_j^L = \frac{\partial E}{\partial I_j^L}$ for a neuron n_j^L and its input I_j^L. Classically, the output layer generally uses a Softmax activation function in order for its activations to represent estimates of the posterior probabilities for each class.

B. Siamese Neural Networks

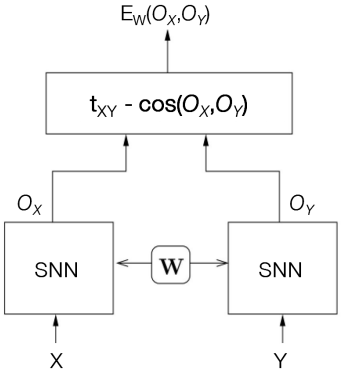

Fig. 1. Architecture of the original Siamese Neural Network

As shown in Figure 1, the principal SNN model was introduced by Bromley *et al.* [2] for signature verification, and applied to face verification by Chopra *et al.* [5]. It contains two feed-forward neural networks with shared weights that, given respectively two input vectors X and Y, structure the output space such that the distance between the two sample outputs O_X and O_Y reflects a semantic similarity. The SNN inherits some MLP characteristics. However, the output layer activations are not considered as posterior probabilities, but as a feature vector.

The SNN error function originally relies on the cosine similarity distance. Let $C = \{C_1, .., C_K\}$ be the set of classes represented in the training data, $O_R(W)$ the output vector of the network for a reference sample x_R from class C_i, O_P the output vector of a second *positive* sample x_P from the same class, and $O_{N_l}, l \neq k$ the output vector of a *negative* sample $x_{N_l}, l \neq k$ from a different class. The goal of the SNN is to maximize inter-class variances while minimizing intra-class variances, meaning O_R and O_P should be collinear while O_R and O_{N_l} should be orthogonal. Consequently, one sample is not enough any longer to define an estimate of the error, and training pairs have to be selected. While Bromley *et al.* in [2] defined separate positive and negative pairs, whose number was arbitrary, Lefebvre *et al.* in [9] proposed an error criterion based on triplets, with one reference example, one negative and one positive examples.

In order to keep symmetric roles for every class and optimize the efficiency of every update, we propose here to minimize an error criterion for training subsets $T = \{x_R, x_P, \{x_{N_l}, l = 1..K, l \neq k\}\}$ involving one reference sample, one positive sample and one negative sample from every other class. The error estimation $E_W(T)$ becomes:

$$E_W(T) = (1 - cos(O_R, O_P))^2 + \sum_l (0 - cos(O_R, O_{N_l}))^2. \quad (4)$$

For numerical stability reasons, we also propose to replace the cosine distance for each pair by a combination of multiple factors. The scalar product $O_1.O_2$ between two sample outputs O_1 and O_2 was used instead of the cosine, and additional constraints are added on the norms of both outputs, forcing them to one. These conditions ensure that the cosine distance between these outputs is still equal to the original target, while preventing any saturation of the outputs, given:

$$cos(O_1, O_2) = \frac{O_1.O_2}{\|O_1\| \, . \, \|O_2\|}. \quad (5)$$

Thus, we define the final error estimation over all the chosen training subsets $T_s, s \in [\![1, \tau]\!]$ E_W as:

$$E_W = \sum_{s \in [\![1,\tau]\!]} E_W(T_s), \quad (6)$$

with

$$\begin{aligned} E_W(T_s) =& (1 - O_R.O_P)^2 + \sum_l (0 - O_R.O_{N_j})^2 \\ &+ \sum_k (1 - \|O_k\|)^2. \end{aligned} \quad (7)$$

The backpropagation algorithm is then modified in order to take into account the part played by all samples. Thus, we define, for a neuron n_i, δ_{R_i}, δ_{P_i}, $\{\delta_{N_{li}}\}$, generalized versions of the δ defined earlier, in relation to their corresponding input sample in the training subset. Given the activations of the neuron n_i for all the samples of a training subset, the error for a weight ω_{ji} can then be computed by the following equation :

$$\frac{\partial E}{\partial \omega_{ji}} = \delta_{R_i} a_{R_i} + \delta_{P_i} a_{P_i} + \sum_l \delta_{N_{li}} a_{N_{li}} \quad (8)$$

Since an SNN is trained to evaluate multiple gesture similarities, our assumption to be experimented is that unknown samples are projected into a feature space in a coherent manner with known classes. This hypothesis is then tested in an SNN rejection strategy, presented in the following paragraph.

C. Rejection strategies

Once the SNN is trained, the output layer gives a feature vector representing a similarity measure of a set of samples. Any classifier can be used on these feature vectors. We choose a K-NN classification based on the cosine similarity metric in order to prove the validity and reliability of the learned SNN projection. Indeed, while the K-NN classifier does not scale efficiently for larger datasets, it stays relevant for the domain of gesture recognition. Finally, our rejection criterion consists in a single threshold, common to all classes, on the distance to the closest known sample. A similar thresholding criterion is also applied to a DTW-based model in order to get a fair comparison and to a MLP rejection strategy based on the maximum posterior probability. Two kinds of rejection are then studied. The first kind encompasses all incorrect classifications, and tests the ability of a system to identify samples whose classification is too uncertain to be accepted. The main challenge for the model is to isolate the misclassified samples first. The second kind of rejection concerns the "rest of the world" paradigm, and aims at evaluating a model performance in isolating elements it was not trained for from the rest of the known classes. This rejection is only rarely taken into account by existing methods, or is taken care of by another model specifically trained for this task.

IV. Experiments

A. Datasets and preprocess

To our knowledge, no public dataset is available for 3D gesture recognition benchmarks to this date. Thus, we collected two datasets, using an Android Samsung Nexus S device, at a sampling rate of 40Hz. The first dataset, DB1, gathers 40 repetitions of 18 different classes performed by a single individual, for a total of 720 records. DB1 is the base for testing personalised models, fitted for a particular user. The second dataset, DB2, gathers 5 repetitions of 14 different classes performed by 22 individuals, for a total of 1540 records. DB2 allows for a more generalized testing, in an open-world with multiple users. The 14 classes of DB2 are formed of linear gestures, with horizontal translations ('flick North, South, East, West') and vertical translations ('flick Up, Down'); curvilinear gestures ('clockwise' and counter-clockwise' circles, 'alpha', 'heart', 'N' and 'Z' letters, 'pick' gesture towards the user, and 'throw' gesture away from the user). The 4 additional classes in DB1 are the number '8', the symbol 'infinity' and the letters 'V' and 'W'.

The accelerometer and gyrometer signals are then preprocessed in 3 steps in order to build a non-temporal vector for MLP and SNN learning. First, amplitude scaling, where each component of every sample (3D accelerometer and 3D gyrometer) forming a gesture record is divided by the maximum norm over all the samples of this gesture, reduces amplitude variations between different gestures dynamics, and ensures that input values are between -1 and +1, which is recommended for an efficient neural network training. Then, a low-pass filter is applied to increase the signal-noise ratio. Finally, gestures are normalized over time by forcing the same fixed size, set to 45 after preliminary experiments, for every gesture. This is done by computing the curvilinear distance of the whole gesture, before linearly interpolating or extrapolating the final samples at fixed coordinates.

Temporal input data are filtered, normalized and vectorized for the implementation of the DTW based method [15] in our comparative protocols.

B. Testing protocols

Two types of rejection are studied in this paper. First, we test the rejection quality for misclassified samples whose outputs are too far from what the network learned. The test protocol P1 is based on DB2: all the records from one individual are used for training, while the other 21 individuals' records are used for testing. This allows for testing the generalization potential of the trained model, as well as its capacity for rejecting samples when variations with the reference individual are too important. It is the most challenging representation of an open-world, where not every user can be taken into account when training the model. Finally, we test the ability for a model to reject unknown gestures. The test protocol P2 is based on DB1. 14 gesture classes are used during the training phase, with 5 repetitions per class. The test data comprises 16 repetitions from each of these classes, as well as every record available from the 4 additional classes, for a total of respectively 224 and 160 records from known and unknown classes. This test embodies a realistic personalization paradigm, used in a natural user interface where the user does not specify when they make a gesture, and the system has to determine whether to trigger an event even before selecting the corresponding event. In our experiments, every protocol is repeated 10 times in order to get meaningful average classification results.

C. SNN Parameter determination

The following meta parameters have to be tuned to optimize the learning sample representation: the learning rate, the number and sizes of the different layers and the number of training sets presented to the model at each epoch. The final parameters were decided to be the same for every configuration to prevent any specific unrealistic tuning to the test data. In the first place, the learning rate was set to a low value of 5.10^{-5} in order to improve the convergence of our modified backpropagation algorithm. Then, after preliminary tests, the number of hidden layers was set to 1, with 45 hidden neurons, which seems coherent with the preprocessed samples of temporal length equal to 45.

Finally, we studied the influence of the size of the output layer, since a higher size corresponds to a higher number of descriptors available for the classifier. In order to evaluate the performances of each model for every protocol, we consider the classification rate relative to the rejection rate applied. The classification rate is defined as the ratio between the number of samples accepted and correctly classified, and the total number of accepted samples; and the rejection rate, as the ratio between the number of rejected samples and

the total number of samples. A higher area under the curve implies a better rejection and classification quality.

Our evaluations show that increasing the SNN output size is beneficial to the network discrimination capacity. A threshold was quickly reached from 10 to 100 for the single-user configuration (cf. Fig. 2.a), whereas an increased size of the output layer was beneficial for the multi-user configuration (cf. Fig. 2.b). The final size was set to 80 for both protocols. In the case of P2, since 41.6% of the samples are unknown in the test dataset, no classifier can achieve a 100% accuracy under a 41.6% rejection rate threshold. This score limitation is depicted with the filled area, and the perfect score lines on 2.b and 3.b.

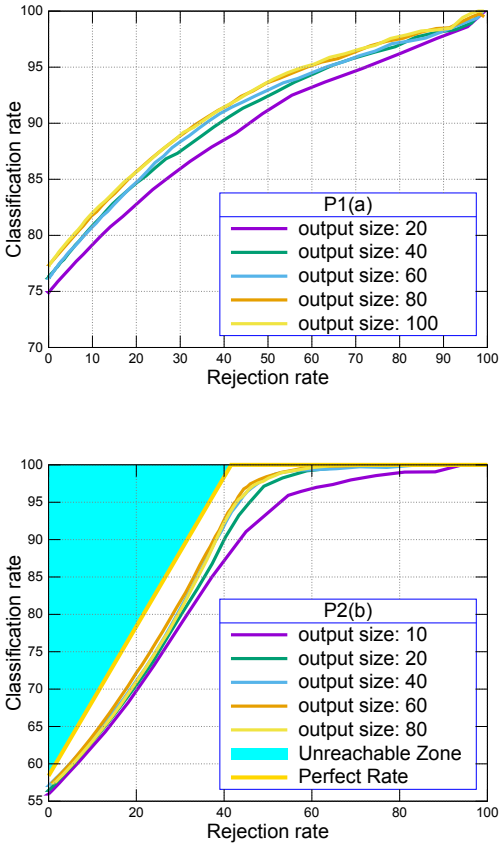

Fig. 2. SNN classification for different outputs size on P1 (a) and P2 (b).

D. Results

In the following, we compare the SNN results obtained with the final parameter configuration to our best DTW and MLP performances. P1 shows that, while the SNN outperforms the MLP for misclassification detection, it is still less efficient than the DTW based model, which can be explained by the lack of data necessary for ensuring the generalization properties of a neural network. For a realistic 10% rejection rate, the DTW correct classification rate is equal to 84%, while the SNN and the MLP get a respective score of 82% and 79% (cf. Fig. 3.a). However, P2 shows the superior capacity of the SNN to isolate unknown samples. Around the 41.6% landmark, where every unknown sample can be rejected, the SNN presents a correct classification rate of 94%, while the DTW and the MLP get lower respective scores of 92% and 88%. Furthermore, in its best configurations, depicted by the means of the deviation, the SNN is the closest to the perfect rate (see the yellow line in Fig. 3.b) as the rejection rate increases.

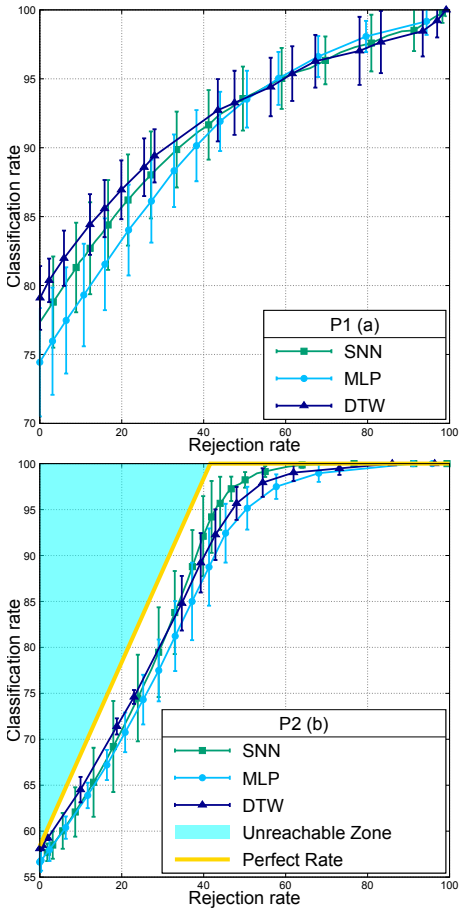

Fig. 3. DTW, MLP, SNN comparison on P1 (a) and P2 (b).

The figure 4 shows the rejection performance for the three tested methods on P2. The evolution of three types of rejection is followed as the rejection rate increases. Rejected misclassifications and samples from unknown classes form the right rejection, while rejected samples which would have been correctly classified form the wrong rejection. It is very interesting to observe then that the SNN-based method presents the lowest area for the mean wrong rejection rate, with a steady right rejection rate that only starts to degrade after the 41.6% landmark, showing its greater selection ability compared to the other two methods where the degradation is a lot more spread with the increase of the rejection rate. The gap with the perfect rejection model, where the area under the perfect rejection line on 4 would be dedicated entirely to unknown classes rejection, is also a lot smaller for the SNN. Thus, we can conclude that our goal to minimize intra-class distances and maximize inter-

class distances is reached, and in a more efficient way than the classical geometric or machine learning approaches.

V. CONCLUSIONS AND FUTURE WORKS

We presented an inertial gesture recognition and rejection approach based on a non-linear similarity metric. Using different datasets in order to cover realistic challenging cases, we showed that the suggested modified SNN proves to be superior for unknown and novel gesture detection to the main similarity based model used in state-of-the-art methods. It also outmatches its neural counterpart, the MLP, both in classification and rejection capabilities. Nevertheless, the DTW based model still outperforms our model for misclassifications rejection, which can be explained by the temporal aspect of the data giving an edge to the DTW. This is the reason why we aim at developing an improved SNN model which will be able to handle time series, simplifying further the preprocess step and increasing the network's discrimination capacity.

REFERENCES

[1] Akl, A, and Valaee, S., Accelerometer-based gesture recognition via dynamic-time warping, affinity propagation, & compressive sensing, in IEEE ICASSP, pp.2270,2273, 14-19 March 2010.

[2] J. Bromley, I. Guyon, Y. Lecun, E. Sckinger, and R. Shah, Signature Verification using a Siamese Time Delay Neural Network, in NIPS Proc, 1994.

[3] Cho, S.-J. and Choi, E. and Bang, W.-C. and Yang, J. and Sohn, J. *et al.* , Two-stage Recognition of Raw Acceleration Signals for 3-D Gesture-Understanding Cell Phones, 10th International Workshop on Frontiers in Handwriting Recognition, 2006.

[4] B. Choe, J.-K. Min, et S.-B. Cho, Online Gesture Recognition for User Interface on Accelerometer Built-in Mobile Phones, in Neural Information Processing. Models and Applications, vol. 6444, Springer, pp. 650-657, 2010.

[5] S. Chopra, R. Hadsell, and Y. LeCun, Learning a similarity metric discriminatively, with application to face verification, in IEEE CVPR, vol. 1, pp. 539-546, 2005.

[6] S. Duffner, S. Berlemont, G. Lefebvre, and C. Garcia, 3D gesture classification with convolutional neural networks, in IEEE ICASSP 2014, pp. 5432-5436, 2014.

[7] S. S. Fels and G. E. Hinton, Glove-Talk: a neural network interface between a data-glove and a speech synthesizer, IEEE Transactions on Neural Networks, vol. 4, n. 1, pp. 2-8, janv. 1993.

[8] A.E. Gasca, T.S. Saldaña, G.S. Sánchez, G.V. Velásquez, L.R. Rendn, B.I. Abundez, R.R. Valdovinos and R.R. Cruz, A rejection option for the multilayer perceptron using hyperplanes, in Adaptive and Natural Computing Algorithms, Springer, pp. 5160, 2011.

[9] G. Lefebvre and C. Garcia, Learning a bag of features based nonlinear metric for facial similarity, in IEEE AVSS, pp. 238-243, 2013.

[10] G. Lefebvre, S. Berlemont, F. Mamalet, and C. Garcia, BLSTM-RNN Based 3D Gesture Classification, in ICANN, Springer, pp. 381-388, 2013.

[11] T. Pylvänäinen, Accelerometer Based Gesture Recognition Using Continuous HMMs, in Pattern Recognition and Image Analysis, Springer, pp. 639-646, 2005.

[12] S. Singh and M. Markou, An approach to novelty detection applied to the classification of image regions, IEEE Transactions on Knowledge and Data Engineering, vol. 16, n. 4, p. 396-407, avr. 2004.

[13] G. C. Vasconcelos, M. C. Fairhurst, and D. L. Bisset, Investigating the recognition of false patterns in backpropagation networks, in ICANN, pp. 133-137, 1993.

[14] J. Wu, G. Pan, D. Zhang, G. Qi and S. Li, Gesture Recognition with a 3-D Accelerometer, Ubiquitous Intelligence and Computing, Springer, pp. 25-38, 2009.

[15] E. Petit. GRASP: Gesture Recognition Engine. Dpt Logiciel, France Télécom, IDDNFR.001.030023.000.S.P.2010.000.31500, 2010

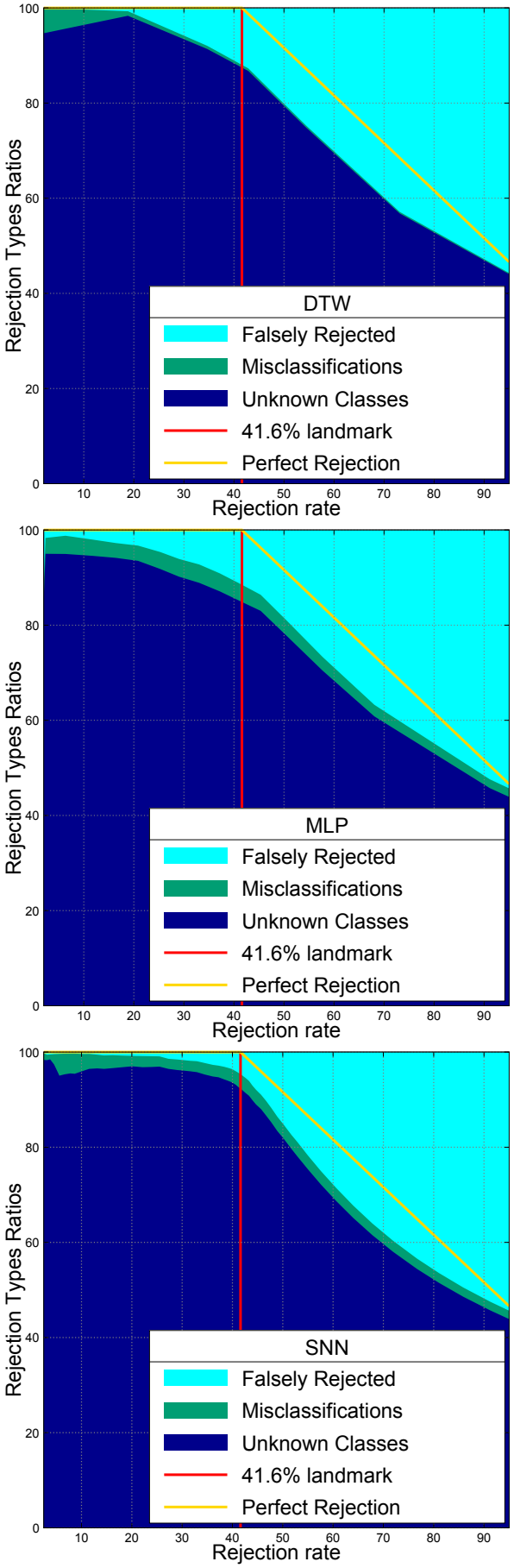

Fig. 4. DTW, MLP, SNN rejection details on P2