Neural Network based Novelty Detection for Incremental Semi-supervised Learning in Multi-class Gesture Recognition

Husam Al-Behadili[1,2], Arne Grumpe[2] and Christian Wöhler[2]
[1]*Electrical Engineering, University of Mustansiriyah, Al-Mustansiriyah, Box 46007, Baghdad, Iraq*
[2]*Image Analysis Group, TU Dortmund University, Otto-Hahn-Str. 4, D–44227, Dortmund, Germany*

Keywords: Data Stream, Neural Network, Extreme learning Machine (ELM), Novelty Detection, Incremental Learning, Semi-supervised Learning, Extreme Value Theory (EVT), Confidence Band.

Abstract: The problems of infinitely long data streams and its concept drift as well as non-linearly separable classes and the possible emergence of "novel classes" are topics of high relevance for gesture data streaming based automatic recognition systems. To address these problems we apply a semi-supervised learning technique using a neural network in combination with an incremental update rule. Neural networks have been shown to handle non-linearly separable data and the incremental update ensures that the parameters of the classifier follow the "concept-drift" without the necessity of an increased training set. Since a semi-supervised learning technique is sensitive to false labels, we apply an outlier detection method based on extreme value theory and confidence band intervals. The proposed algorithm uses the extreme learning machine, which is easily updated and works with multi-classes. A comparison with an auto-encoder neural network shows that the proposed algorithm has superior properties. Especially, the processing time is greatly reduced.

1 INTRODUCTION

When people start a conversation, they commonly use hand motions. These motions of the hand may also be used to communicate with machines in Human-Machine Interaction (HMI). In this context, they are called "gestures". Since gestures are considered an intuitive, fast and save way in HMI, they are included in many applications ranging from computer games to applications in the medical industry (e.g. Yusoff et al. (2013)). Gestures of the same class, however, may be performed in a different manner by different persons or by the same individual if she/he performs the gesture more than once. Hence, the classifier should be trained with all possible gestures to get an acceptable recognition rate. Unfortunately, manually labelled data are scarce. Unlabelled data, in contrast, may be streamed continuously. Consequently, semi-supervised learning may be applied to solve this problem. Semi-supervised learning corresponds to first training a classifier on a labelled data set in a supervised manner and updating the training set using the labels assigned by the classifier itself (Zhu and Goldberg, 2009). However, the problems arising from the usage of streamed data are the "infinite length" and the "concept-drift". They were addressed by most of the existing on-line algorithms (Masud et al., 2011). In addition, non-linearly separable distributions of the data, the emersion of new classes or outliers, and the computational complexity are possible problems. To address these problems we apply a semi-supervised learning technique using a neural network in combination with an incremental update rule. Neural networks have been shown to handle non-linearly separable data and the incremental update ensures that the parameters of the classifier follow the "concept-drift" without the necessity of an increased training set. Since a semi-supervised learning technique is sensitive to false labels, we apply an outlier detection method based on the concepts of extreme value theory and confidence bands interval to suppress false labels that will potentially affect the performance of the classifier after the next training cycle.

2 RELATED WORK

There have been several research studies on semi-supervised learning (Zhu and Goldberg, 2009). The semi-supervised techniques are categorized in three main types, which are: First, guessing the unlabelled data and subsequently retraining the classifier using

Al-Behadili, H., Grumpe, A. and Wöhler, C.
Neural Network based Novelty Detection for Incremental Semi-supervised Learning in Multi-class Gesture Recognition.
DOI: 10.5220/0005674202870294
In *Proceedings of the 11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2016) - Volume 3: VISAPP*, pages 289-296
ISBN: 978-989-758-175-5

these labels, e.g. EM, co-training, and transductive SVM (Zhu and Goldberg, 2009). Second, finding additional features derived from the unlabelled data (Johnson and Zhang, 2015). Third, manifold regularization has been used to leverage the labelled and unlabelled data (Huang et al., 2014). Recently, the extreme learning machine was applied to semi-supervised learning, since it has favourable features. Huang et al. (2014) constructed the Laplacian graph from labelled and unlabelled data to extend the extreme learning machine for semi-supervised learning. They, however, supposed all unlabelled data to be available together at the beginning of the training. Hence, it is not suitable for data streams. A new approach by Li et al. (2013) applied co-training to train the ELM in a semi-supervised manner. Since, this algorithm repeatedly trains several ELMs, it is considered computationally costly. Since neural networks have the ability to approximate the non-linear mapping from features to classes directly from the input samples, several researchers proposed different algorithms of incremental neural networks. Huang et al. (2006a) introduced the "incremental extreme learning machine" (IELM) by adding randomly generated nodes to a single-layer feed-forward network (SLFN) and computed the output weight analytically for the new nodes only. Since semi-supervised learning is sensitive to false labels, the proposed algorithm needs to detect and reject the outliers which may affect on the performance of the algorithm. Pimentel et al. (2014) describe methods solving the novelty detection problem by using statistical techniques or neural networks. Applications of neural networks have been implemented in many domains (Hugueny et al., 2009). Due to the large variety of artificial neural networks many different neural network approaches may be used for novelty detection. Tax (2001) used the auto-associative neural networks (AANN) as an one class classifiers within the data discretion, outlier and novelty detection toolbox (ddtools)[1] (Tax, 2015), where the auto-encoder function is called "autoenc_dd". It is a reference for our work and is explained in detail in the next section.

3 REFERENCE METHODS

3.1 Extreme Learning Machine (ELM)

The concept of the ELM is described in detail by Huang et al. (2006b), whose description we follow here. According to their approach, for a SLFN with L hidden neurons the output is given by

$$f_l(\vec{x}_j) = \sum_{i=1}^{L} \beta_i G(\vec{a}_i, b_i, \vec{x}_j) \quad \vec{x}_j \in \mathcal{R}^n, \vec{a}_i \in \mathcal{R}^n \quad (1)$$

with β_i as the output weight, $\vec{a}_i$ and b_i as the learning parameters, $G(\vec{a}_i, b_i, \vec{x}_j)$ as hidden neuron output i, and $\vec{x}_j$ as the feature vector associated with training sample j. It is

$$G(\vec{a}_i, b_i, \vec{x}_j) = g(\vec{a}_i \cdot \vec{x}_j + b_i), \quad b_i \in \mathcal{R} \quad (2)$$

with $\vec{a}_i$ and b_i as the ith neuron's input weight vector and bias. For hidden neurons with radial basis function (RBF) characteristic it is,

$$G(\vec{a}_i, b_i, \vec{x}_j) = g(b_i \left\| \vec{x}_j - \vec{a}_i \right\|) \quad b_i \in \mathcal{R}^+ \quad (3)$$

with $\vec{a}_i$ and b_i as the centre and width of RBF neuron i. $\mathcal{R}^+$ refers to all positive real numbers (Huang et al., 2006b).

The ELM is a SLFN network, and the equations above apply to it. Following Huang et al. (2006b), suppose we have m arbitrary distinct sample ($\vec{x}_j \in \mathcal{R}^n$, $\vec{t}_j \in \mathcal{R}^c$) consisting of a feature vector $\vec{x}_j$ and a target vector $\vec{t}_j$ containing one value for each of the c classes, respectively. Notably, it is $\vec{t}_j = +1$ for the output neuron belonging to the class of the sample and $\vec{t}_j = -1$ for the other output neurons, respectively. The ELM has L neurons. An error-free approximation of the m samples by this ELM then implies the existence of a set of parameters β_i, $\vec{a}_i$ and b_i fulfilling

$$f_L(\vec{x}_j) = t_j, j = 1 \cdots m, \quad (4)$$

which can be written as a matrix

$$\boldsymbol{H} \cdot \boldsymbol{\beta} = \boldsymbol{T} \quad (5)$$

$$\boldsymbol{H} = \begin{bmatrix} G(\vec{a}_1, b_1, \vec{x}_1) & \cdots & G(\vec{a}_L, b_L, \vec{x}_1) \\ \vdots & \ddots & \vdots \\ G(\vec{a}_1, b_1, \vec{x}_m) & \cdots & G(\vec{a}_L, b_L, \vec{x}_m) \end{bmatrix}_{m \times L}, \quad (6)$$

$$\boldsymbol{\beta} = \begin{bmatrix} \vec{\beta}_1^T \\ \vdots \\ \vec{\beta}_L^T \end{bmatrix}_{L \times c} \text{ and } \boldsymbol{T} = \begin{bmatrix} \vec{t}_1^T \\ \vdots \\ \vec{t}_m^T \end{bmatrix}_{m \times c} \quad (7)$$

where $\vec{\beta}_i$ denotes the vector containing the ith neuron's output weight for all classes and the mtrix $\boldsymbol{H}$ denotes the hidden layer output. According to Huang et al. (2006b), the procedure of training the ELM is as follows:

- The first step is to assign the input parameters (i.e. $\vec{a}_i$, and b_i, $i = 1, \ldots, L$) randomly.
- Analytical computation of the matrix $\boldsymbol{H}$ is performed according to Eq. (7).

[1] http://prlab.tudelft.nl/david-tax/dd_tools.html

- The output weights are then estimated using (5).

As shown by Huang et al. (2006b), the problem here is to minimize the error in (5), i.e. $\|\boldsymbol{H}\cdot\hat{\boldsymbol{\beta}}-\boldsymbol{T}\|$. Since (5) is a linear system in the output weights, the output weights are estimated by Huang et al. (2006b) using the pseudoinverse of the output matrix of the hidden layer according to $\boldsymbol{H}^{\dagger}=(\boldsymbol{H}^{T}\boldsymbol{H})^{-1}\boldsymbol{H}^{T}$ (Rao and Mitra, 1971):

$$\hat{\boldsymbol{\beta}}=\boldsymbol{H}^{\dagger}\cdot\boldsymbol{T} \tag{8}$$

As suggested by Huang et al. (2006b), the singular value decomposition (SVD) (Rao and Mitra, 1971) is used to compute the pseudoinverse $\boldsymbol{H}^{\dagger}$. The labels of the new samples can then be obtained by using the estimates $\hat{\boldsymbol{\beta}}$ and $\boldsymbol{H}^{\dagger}$ in (7).

3.2 Extreme Value Theory (EVT)

The conventional approaches of the novelty detection mostly require one threshold or a set of class-wise thresholds computed by additional manually labelled data or by using the cross-validation process, which are either need additional labelled data or are time consuming. In contrast, the extreme value theory (EVT) as described by Roberts (1999) and Clifton et al. (2008) avoids these problems. It is a statistical theory used to model the extreme values, i.e. maxima or minima in the one-dimensional case, in the tails of the distributions. Following Roberts (1999), let there be a set of m i.i.d. random samples $\mathcal{X}=\{\vec{x}_1,\vec{x}_2...,\vec{x}_m\}$ each of them is n-dimensional and distributed as given by the probability density function $f(\vec{x})$. Furthermore, let the extreme value of $\mathcal{X}$ be $\vec{x}_{\max}$. Given the set $\mathcal{X}$, the probability of $\vec{x}$ being more extreme than $\vec{x}_{\max}$ is the extreme value probability which expressed as $P_{\mathrm{EV}}(\vec{x}|\mathcal{X})=P(\vec{x}_{\mathrm{x}}<\vec{x})$. There are three types of distributions of the EVD as stated by the Fisher-Tippett theorem (Fisher and Tippett, 1928): the Gumbel distribution, the Frechet distribution and the Weibull distribution (Roberts, 1999). As described by Roberts (1999) and Clifton et al. (2008), the Gumbel distribution models the EVD of data originating from the one-dimensional one-sided normal distribution with a mean value of zero and a variance of 1, i.e. $D=\left|\mathcal{N}(0,1)\right|$. According to Clifton et al. (2008), the Gumbel distribution for a one-dimensional variable x

$$P_{\mathrm{Gumbel}}(x|\mathcal{X})=\exp\left[-\exp\left(-\frac{x-\mu_m}{\sigma_m}\right)\right], \tag{9}$$

is defined by the location parameter

$$\mu_m=(2\ln(m))^{0.5}-\frac{\ln(\ln(m))+\ln(2\pi)}{2(2\ln(m))^{0.5}} \tag{10}$$

and the scale parameter

$$\sigma_m=(2\ln(m))^{0.5}. \tag{11}$$

Both parameters depend only on m, i.e. the number of samples drawn from D. These minimize or absorb the effect of the amount of training data on the final results and thus the threshold remains unchanged with an increasing amount training data.

We define a threshold P_{th} that is based on the extreme values of the known classes. A novelty is detected if P_{EV} exceeds this threshold. As may be verified from the equations above, the threshold in EVT has a direct statistical interpretation and does not depend on the distribution of the classes whereas the conventional thresholds depend on the distribution of the classes.

3.3 Auto-associative Neural Networks

Auto-associative neural networks (AANN) or auto-encoder networks (AutoENC), as they are called by Tax (2001), are neural networks that learn a data representation (Hertz et al., 1991), i.e. an AANN reconstructs the input pattern at their output layer. In this work, we apply the auto-encoder from the toolbox presented by Tax and Duin (1999). In this toolbox the auto-encoder architecture has only one hidden layer with h_{auto} hidden units. Sigmoid transfer functions are used for the hidden neurons. It is trained by minimizing the mean squared deviation of the input from the output. The error is used as a measure for the novelty detection. It is supposed that the target patterns will be reconstructed with smaller errors than outliers. The error E_{AANN} of an input $\vec{x}$ is

$$E_{\mathrm{AANN}}=\|f_{\mathrm{AANN}}(\vec{x},\vec{w})-\vec{x}\|^2 \tag{12}$$

where f_{ANN} is transfer function of the AANN and $\vec{w}$ is a vector containing its parameters. The problems of the AANN to novelty detection are the same problems as those arising from the conventional application of neural networks to classification problems: It requires a pre-defined number of neurons, a learning rate, and stopping criteria from an expert user. In our experiments, the best number of neurons is 5, which is the default number in the function of the toolbox. We also used the default outlier percentage 5% that is used to compute the threshold of the novelty.

3.4 Confidence Bands

Since measurements are affected by noise, models derived from these measurements differ for each acquired set of new data. The confidence band encloses all models obtained from the measurements with a specified probability (Kardaun, 2005). Confidence bands are computed in several approaches

(e.g. Kendall et al., 2007). In the context of semi-supervised learning, the confidence bands of a polynomial classifier are used by Al-Behadili et al. (2014) to detect outlier samples.

4 THE PROPOSED ALGORITHM

Here, we propose a method to update the ELM incrementally and to apply the outliers detection using the EVT and the confidence band to the output of the ELM to reject the outliers. The proposed algorithm extends the approach of Al-Behadili et al. (2015), which uses only EVT for detecting the outliers. It consists of two phases.

4.1 Incremental Learning Phase

The incremental updating rule is derived based on the pseudoinverse method introduced by Lan et al. (2009) according to

$$\boldsymbol{M} = \boldsymbol{H}^T\boldsymbol{H} \text{ and } \boldsymbol{P} = \boldsymbol{H}^T\boldsymbol{T}. \quad (13)$$

$$\text{Hence,} \quad \boldsymbol{\beta} = \boldsymbol{M}^{-1}\boldsymbol{P}. \quad (14)$$

The dimension of $\boldsymbol{M}$ is $L \times L$ and the dimension of the $\boldsymbol{P}$ is $L \times c$, where L corresponds to the number of hidden neurons and c to the number of classes.

Suppose that we have $m_{(0)}$ labelled samples for initial training. We then compute $\boldsymbol{M}_{(0)} = \boldsymbol{H}_{(0)}^T\boldsymbol{H}_{(0)}$ and $\boldsymbol{P}_{(0)} = \boldsymbol{H}_{(0)}^T\boldsymbol{T}_{(0)}$ according to (13). Hence, $\boldsymbol{\beta}_{(0)} = \boldsymbol{M}_{(0)}^{-1}\boldsymbol{P}_{(0)}$.

Incremental learning is achieved by adding chunks of samples to the training set. If the number of samples $\vec{x}$ in the chunk $k+1$ is $\hat{m}$ then the hidden layer output matrix $\hat{\boldsymbol{H}}$ corresponding to the new chunk of data is

$$\hat{\boldsymbol{H}} = \begin{bmatrix} G(\vec{a}_1, b_1, \vec{\hat{x}}_1) & \cdots & G(\vec{a}_L, b_L, \vec{\hat{x}}_1) \\ \vdots & \ddots & \vdots \\ G(\vec{a}_1, b_1, \vec{\hat{x}}_{\hat{m}}) & \cdots & G(\vec{a}_L, b_L, \vec{\hat{x}}_{\hat{m}}) \end{bmatrix}_{\hat{m}\times L} \quad (15)$$

From (15) and (13) it follows that $\hat{\boldsymbol{M}} = \hat{\boldsymbol{H}}^T\hat{\boldsymbol{H}}$ and $\hat{\boldsymbol{P}} = \hat{\boldsymbol{H}}^T\hat{\boldsymbol{T}}$ corresponding to the new chunk data. Then

$$\boldsymbol{M}_{(k+1)} = \boldsymbol{M}_{(k)} + \hat{\boldsymbol{M}} \text{ and } \boldsymbol{P}_{(k+1)} = \boldsymbol{P}_{(k)} + \hat{\boldsymbol{P}} \quad (16)$$

Finally, using (14) the updated output is $\boldsymbol{\beta}_{(k+1)} = \boldsymbol{M}_{(k+1)}^{-1}\boldsymbol{P}_{(k+1)}$.

4.2 Novelty Detection Phase

4.2.1 Novelty Detection using EVT

According to Huang et al. (2006b), the output of the ELM is around $+1$ for the class that the sample belongs to it and around -1 for the other classes. This follows immediately from the target values used for the training of the ELM. If the output of the winner class is exactly $+1$ then this result is similar to the training data and thus highly believable. The confidence of the result decreases with an increasing distance of the winner output class from the ideal value of $+1$. Furthermore, the linear least squares optimization applied in the training yields mean-free normally distributed residuals. Consequently, the absolute difference between the ELM output and the ideal value, i.e. a vector that contains $+1$ at the position of the winning neuron and -1 at all other positions, will originate from a mean free one-sided normal distribution. Additionally, we divide the absolute difference by the standard deviation of the residuals, i.e. the sum of the squared residuals, to arrive at a $\mathcal{N}(0,1)$ distribution.

Recalling (5) and substituting (8), we arrive at the prediction $\boldsymbol{D}$ of the training set

$$\boldsymbol{D} = \boldsymbol{H}(\boldsymbol{H}^T\boldsymbol{H})^{-1}\boldsymbol{H}^T\boldsymbol{T}. \quad (17)$$

Notably, each column of $\boldsymbol{D}$ contains the predicted values for one class. Let $\vec{d}_c$ and $\vec{t}_c$ be the vector containing the predicted values and the target values of class c, respectively. The sum of the squared residuals is then given by

$$r_c = \left[\vec{d}_c - \vec{t}_c\right]^T \left[\vec{d}_c - \vec{t}_c\right] \quad (18)$$

$$= \vec{t}_c^T\boldsymbol{H}(\boldsymbol{H}^T\boldsymbol{H})^{-1}\boldsymbol{H}^T\vec{t}_c + \vec{t}_c^T\vec{t}_c. \quad (19)$$

Notably, $\boldsymbol{H}^T\boldsymbol{H} = \boldsymbol{M}$ and $\boldsymbol{H}^T\vec{t}_c$ is the cth column of $\boldsymbol{P}$. Consequently, the first summand on the right side of (19) may be incremented using (16). The term $\vec{t}_c^T\vec{t}_c$ is the sum of the squared target values. Consequently, it may be incremented by adding the squared target values of additional samples. Dividing r_c by the number of samples and taking the square root results in the standard deviation of the residuals. Notably, this approach yields a class-wise standard deviation which represents the different models formed by the output layer of the ELM.

After division by the class-wise standard deviation, the absolute difference of the ELM output and the ideal value originates from a one-sided normal distribution and is thus modelled by the Gumbel distribution of the extreme value theory (Clifton et al., 2008). Accordingly, the highly believable samples have $P_{ev} = 0$ and the ideal novel sample yields $P_{ev} = 1$. Thus we set $P_{\text{th}} = 0.9$ and flag a sample to be novel if at least two output neurons detect a novelty.

To ensure only trusted labels in the new training set, we apply additional conditions. If at least one output neuron detects a novelty, we do not consider the

label trustworthy and do not add it to the training set. Furthermore, since in the ideal case the winner class output value should be $+1$ and all other neurons output -1 and the difference between the winning neuron and the second largest output value is supposed to be 2. We have noticed that the ELM outputs two positive values in case of unseen classes, resulting in a difference less than 1. Therefore, the newly labelled samples should fulfill another condition to be accepted in the next training phase: The difference between the first and second largest output values should exceed specific threshold. Here, we set the threshold to 1.

4.2.2 Novelty Detection using Confidence Bands

Equation (5) is a system of linear equations for the output weights, i.e. the output of the ELM is a weighted linear combination of the hidden layer activations, where the weights represent the parameters of a linear model (Huang et al., 2006b). The confidence band intervals of the output decision can thus be estimated. Based on the general derivation of the confidence bands of a linear multivariate regression function given by Kardaun (2005), the confidence band $\eta(\vec{q})$ of the neural network output $\vec{q}$ for a test sample $\vec{x}$ can be estimated according to

$$\eta(\vec{q}) = t_{v,\alpha}\sqrt{\vec{q}^T(\boldsymbol{H}^T\boldsymbol{H})^{-1}\vec{q}}\sqrt{\Sigma_i^m r_i^2/v}, \qquad (20)$$

where $r_i = d(\vec{x}_i) - t(\vec{x}_i)$ is the residual of sample i, $v = m - N_p$ is the number of degrees of freedom, with N_p as the number of free parameters of the model, $t_{v,\alpha}$ is the critical value of the t-distribution which depends on v and the probability threshold α. A similar expression is obtained by Al-Behadili et al. (2014) for the confidence band of a polynomial classifier, where the mathematical framework described by Kardaun (2005) is used as well. Here, the number of free parameters N_p is equal to the hidden neuron number. Since $\boldsymbol{H}^T\boldsymbol{H} = \boldsymbol{M}$ and the residuals $r_i = d(\vec{x}_i) - t(\vec{x}_i)$ appear in the squared sum, we apply Eq. (16) and Eq. (19) to incrementally update the confidence bands. The standard value 0.05 of α is used.

The sample $\vec{x}$ is taken to be novel if the inequality

$$d_1(\vec{x}) - d_2(\vec{x}) < z \cdot [\eta_1(\vec{x}) + \eta_2(\vec{x})] \qquad (21)$$

is fulfilled, where $d_1(\vec{x})$ and $d_2(\vec{x})$ are the largest and second largest decision values of the classifier of the sample $\vec{x}$, $\eta_1(\vec{x})$ and $\eta_2(\vec{x})$ the corresponding confidence band widths and z is a given constant (here $z = 75$). The condition (21) has been proposed by Sakic (2012), who used it for the identification of unreliable sample labels in the context of semi-supervised learning.

Finally, the sample $\vec{x}$ has been considered as novel if both conditions $P_{\text{ev}} \geq P_{\text{th}}$ and Eq. (21), which correspond to the EVT and confidence band, respectively, indicate it as novel.

5 GESTURE DATA SET

The database by Richarz and Fink (2011) comprises emblematic gestures of single-hand gestures acquired by a pair of asynchronous stereo cameras used to compute the 3D trajectories. A data set which comprises 3D trajectories performed with a single hand acquired by a Kinect sensor is described by Al-Behadili et al. (2014). Using that data set[2], we segment the original three repetitions per gesture into single repetitions, yielding a total number of 2878 gestures. In addition, we adopt six features from Al-Behadili et al. (2014): the mean and extent in x, y and z direction, respectively. The remaining two features are modified. The seventh feature is a code which is extracted from the direction of movement:

- The principal components of the 3D trajectory are computed and analysed. Let λ_1 and λ_2 be the largest and the second largest eigenvalues of the covariance matrix of the 3D coordinates, respectively. If $\lambda_2 > 0.6\ \lambda_1$ the gesture is considered a two-axis gesture. Otherwise the gesture is considered a one axis gesture. In the former case we keep the first two principal components and in the latter case we keep only the first principal component for the remaining analysis.
- The 3D coordinates are projected onto the selected principal components and the sign of the velocity of each projected coordinate is computed.
- Based on the amount of positive and negative values, we assign the following value to each principal component, respectively. We assign a value of 1 or 2 if more than 80% of the coordinates are positive or negative, respectively. Otherwise, the gesture has no predominant direction and is assigned a value of 3. Furthermore, a value of 0 is assigned to principal components that were not selected in the first step.
- The three direction values are then interpreted as a base-4 number, and the corresponding decimal representation is computed to combine all directions in one numerical value.

Finally, the last feature represents the total length of the normalised gesture. This feature set has been cho-

[2]The complete data set is available at http://www.bv.e-technik.tu-dortmund.de

sen after many experiments with other, more extensive feature sets. Furthermore, a compact feature set is desirable in the context of online learning.

6 EXPERIMENTAL SET-UP

The normal ELM neural network has been proven to be a fast neural network (Huang et al., 2006b). Moreover, the incremental ELM is faster than normal ELM. Hence, the main comparison will focus on the accuracy rather than the time of processing. To show the additional features of the proposed algorithm we compare its results with the auto-encoder neural network in the PRToolbox[3] (Tax and Duin, 1999). Similar to our algorithm, this auto-encoder algorithm has the ability to detect outliers. Hence, we used it in the semi-supervised process to compare the two algorithms.

The 2878 samples of the nine classes are randomly divided into three disjoint data sets with fractions 40%, 40%, and 20% for the training, the learning and the test set, respectively. The training set contains all nine classes and each class is divided separately, i.e. the training set contains 40% of the samples of all classes. The novel class is then introduced by excluding one class from the initial training set. The learning set is split into chunks, so-called "buckets", of 100 samples.

Both classifiers are trained on the initial training set. Then the accuracy and other measures are evaluated based on the test set. Since the learning set emulates the data stream, it is presented to the classifier in buckets, i.e. subsequent chunks of data. The buckets are labelled by each classifier. The data in the bucket, which is considered an "outlier" or not believable, is then removed and the classifiers are updated based on the remaining data, i.e. we modify the training set by adding the remaining samples, and update both classifiers. The process is then repeated for the next bucket of samples. Since the auto-encoder algorithm is not incremental, the algorithm is retrained using the modified training set while we use the proposed incremental update rule for the ELM. The procedure is repeated until all buckets have been presented to the classifiers. The modification of the training set is done in two steps. First, the sample is tested for novelty and then the sample is tested for trusted predictions. We introduce two flags for the requirements to the proposed ELM algorithm. New samples are considered novel if the first flag set, i.e. the EVT output exceed the threshold of 0.9. The selection of the samples which are included into the training set is controlled by the second flag. This second flag is set if the difference between the winner class and the second class exceeds 1.

The auto-encoder labels the new sample with the winner class label or as an "outlier". Originally, the auto-encoder is an one-class classifier. However, using the function "multic" in the toolbox by Tax (2015) allows for the classification of multiple classes. This is achieved by training one classifier for each class. Each classifier outputs a real number between 0 and 1 similar to a probability. If this number is less than a predefined threshold, which was set by selecting the ratio of the outliers in the training set to be 5%, it indicates the new sample as an "outlier". Otherwise, the new sample is labelled as "target". The "multic" function labels the new sample as "outlier" if it does not match any class, i.e. if it is indicated as "outlier" by all classifiers, and it labels the new sample with the most probable class, i.e. the maximum output class, if more than one class is labelled as "target".

Due to the different novelty detection approaches, each algorithm will indicate different outliers. Furthermore, the classifiers may assign different labels to each sample. Consequently, the classification problem solved by each classifier may change after each bucket of samples from the data stream has been analysed. It is thus possible that the training sets of the different classifiers diverge, i.e. contain different samples and possibly false labels. The corresponding changes in the classifier architecture also have a severe influence on the run time. The whole procedure is thus repeated for 100 random subdivisions of the data per class, i.e. each class is omitted from the training set once. We enforce identical random permutations for both classifiers, respectively, during each of the 100 runs.[4]

In addition to the accuracy, we track some novelty detection metrics and the time required for the update/retrain. We used the metric proposed by Masud et al. (2011) to evaluate both algorithms with respect to their rates of novelty detection and classification errors. The novelty detection metrics are M_{new}, F_{new}, and E_{total}, which represent the fraction of novel class samples that are incorrectly classified as existing classes, the fraction of samples belonging to existing classes that are mistaken as belonging to a novel class, and the total misclassification rate. Following Masud et al. (2011), these metrics can be computed using

$$M_{\text{new}} = F_{\text{n}}/N_{\text{c}} \cdot 100 \tag{22}$$

$$F_{\text{new}} = F_{\text{p}}/(N - N_{\text{c}}) \cdot 100 \tag{23}$$

$$E_{\text{total}} = (F_{\text{p}} + F_{\text{n}} + F_{\text{e}})/N \cdot 100 \tag{24}$$

[3]PRTool is available at http://prtools.org/software/

[4]The results of the individual runs are available at http://www.bv.e-technik.tu-dortmund.de

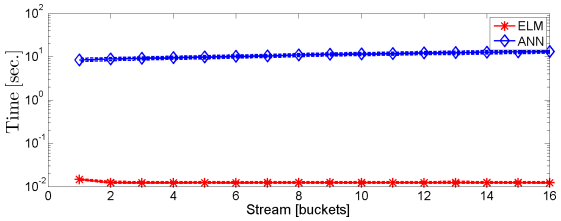

Figure 1: The time required for each bucket, i.e. classification and retraining of both classifiers.

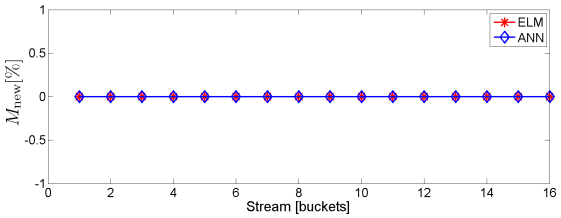

Figure 2: The rate of missed novelties for the incremental ELM and the AANN, respectively.

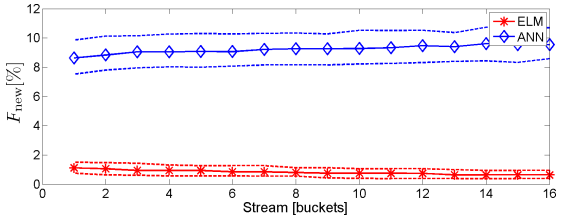

Figure 3: The rate of falsely detected novelties for the incremental ELM and the AANN, respectively.

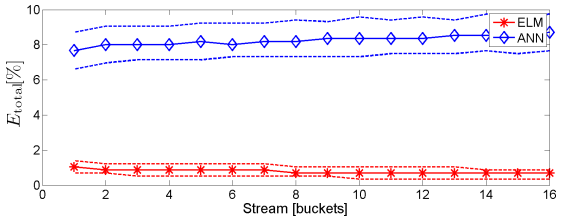

Figure 4: The total error rate for the incremental ELM and the AANN, respectively.

F_{n} is the number of novel class samples that the classifier fails to detect and falsely labels them as existing classes. It corresponds to the number of false negatives for one class classifiers. N_{c} represents the number of samples which belong to the novel class within the presented samples. F_{p} is the number of existing class samples which are wrongly indicated as outliers by the classifier. It corresponds to the number of false positives for one class classifiers. The total number of samples presented to the classifier is denoted by N. F_{e} is the number of existing class samples that are misclassified as other existing classes. As seen from (24), it is not necessary that E_{total} corresponds to the sum of M_{new} and F_{new} (Masud et al., 2011).

7 RESULTS

The run time of the classifier matches the expectations (Fig. 1). Since the proposed algorithm is incremental, it requires less time to adapt. In any case the processing time of the incremental ELM is of the order of some milliseconds which helps to apply it to online data streams.

Fig. 2 shows that the values M_{new} of both algorithms are zero, i.e. no outlier or novelty has been missed. This is important since the outliers are supposed to be near the boundary of the sample distributions of all classes and thus accepting them would significantly affect the performance of the classifiers in the next iterations. Fig. 3 shows the value of F_{new}, which is initially below 2 % for the proposed algorithm whereas it starts at more than 8 % for the AANN. The rate of false detections by the incremental ELM decreases with an increasing amount of training data, reaching a final level of less than 1 %. In the case of the AANN, F_{new} is increasing to more than 9%. Although this means that a small fraction of about 1–2 % of the samples belonging to known classes is supposed to be outliers and, consequently, they are rejected, this is not critical. In fact, this removes the 1–2 % most extreme samples from the semi-supervised training set and thus prevents possibly false labels or sloppily performed gestures from entering the training set. This leads to a slow gradual adaptation of the learned sample distributions, leading to a final stabilised value reflected by E_{total} (Fig. 4). This behaviour is favourable in slowly concept-drifting data streams where the sample distributions change slowly over time. The auto-encoder, in contrast, rejects more samples, which leads to a very slow adaptation. The effect of this novelty detection step is directly apparent in the total error E_{total} of both classifiers. Initially, the error of the proposed approach is around 1 % and decreases to less than 1 %. On the other side, the total error of the auto-encoder is initially around 8 % and increases with an increasing F_{new}.

8 SUMMARY AND CONCLUSION

We have presented an incremental neural network in a semi-supervised learning scenario. In particular, we have applied it to data streaming of emblematic arm gestures, where it is possible that new classes appear based on the continuously streamed data. This relaxes the need for a large and costly manually labelled

data set. Using EVT, the algorithm uses a class-wise statistical threshold for the rejection of outliers. It thus does not require additional labelled data to derive thresholds. More important, it works with multiple classes, and the normalisation prior to the EVT ensures a class-wise threshold. Additionally, it is able to separate the linearly unseparable gesture data. Improvements in the accuracy and processing time are expected when applying this method to other types of data. It might also be helpful for online fault detection in industrial production processes.

REFERENCES

Al-Behadili, H., Wöhler, C., and Grumpe, A. (2014). Semi-supervised learning of emblematic gestures. *At-Automatisierungstechnik*, 62(10):732–739.

Al-Behadili, H., Wöhler, C., and Grumpe, A. (2015). Extreme learning machine based novelty detection for incremental semi-supervised learning. In *3'rd International conference on image Information Processing (ICIIP)*, page In Press. IEEE.

Clifton, D. A., Clifton, L. A., Bannister, P. R., and Tarassenko, L. (2008). Automated novelty detection in industrial systems. In *Advances of Computational Intelligence in Industrial Systems*, pages 269–296. Springer.

Fisher, R. A. and Tippett, L. H. C. (1928). Limiting forms of the frequency distribution of the largest or smallest member of a sample. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 24, pages 180–190. Cambridge Univ Press.

Hertz, J., Krogh, A., and Palmer, R. G. (1991). *Introduction to the theory of neural computation*, volume 1. Basic Books.

Huang, G., Song, S., Gupta, J., and Wu, C. (2014). Semi-supervised and unsupervised extreme learning machines. *IEEE Transactions on Cybernetics*, 44(12):2405–2417.

Huang, G.-B., Chen, L., and Siew, C.-K. (2006a). Universal approximation using incremental constructive feedforward networks with random hidden nodes. *IEEE Transactions on Neural Networks*, 17(4):879–892.

Huang, G.-B., Zhu, Q.-Y., and Siew, C.-K. (2006b). Extreme learning machine: theory and applications. *Neurocomputing*, 70(1):489–501.

Hugueny, S., Clifton, D. A., and Tarassenko, L. (2009). Novelty detection with multivariate extreme value theory, part ii: An analytical approach to unimodal estimation. In *Proc. MLSP*, pages 1–6. IEEE.

Johnson, R. and Zhang, T. (2015). Semi-supervised learning with multi-view embedding: Theory and application with convolutional neural networks. *Proc. CoRR*.

Kardaun, O. J. (2005). *Classical methods of statistics: with applications in fusion-oriented plasma physics*, volume 1. Springer Science & Business Media.

Kendall, W. S., Marin, J.-M., and Robert, C. P. (2007). Confidence bands for brownian motion and applications to monte carlo simulation. *Statistics and Computing*, 17(1):1–10.

Lan, Y., Soh, Y. C., and Huang, G.-B. (2009). Ensemble of online sequential extreme learning machine. *Neurocomputing*, 72(13):3391–3395.

Li, K., Zhang, J., Xu, H., Luo, S., and Li, H. (2013). A semi-supervised extreme learning machine method based on co-training. *Journal of Computational Information Systems*, 9(1):207–214.

Masud, M. M., Gao, J., Khan, L., Han, J., and Thuraisingham, B. (2011). Classification and novel class detection in concept-drifting data streams under time constraints. *IEEE Transactions on Knowledge and Data Engineering*, 23(6):859–874.

Pimentel, M., Clifton, D., Clifton, L., and Tarassenko, L. (2014). A review of novelty detection. *Signal Processing*, 99:215–249.

Rao, C. R. and Mitra, S. K. (1971). *Generalized inverse of matrices and its applications*, volume 7. Wiley New York.

Richarz, J. and Fink, G. A. (2011). Visual recognition of 3d emblematic gestures in an hmm framework. *Journal of Ambient Intelligence and Smart Environments*, 3(3):193–211.

Roberts, S. J. (1999). Novelty detection using extreme value statistics. *IEE Proceedings-Vision, Image and Signal Processing*, 146(3):124–129.

Sakic, D. (2012). Semi-supervised learning using ensemble methods gestures recognition. Master's thesis, University of Dortmund.

Tax, D. (2001). *One-class classification: concept-learning in the absence of counter-examples*. PhD thesis, TU Delft, Delft University of Technology.

Tax, D. (2015). Ddtools, the data description toolbox for matlab.

Tax, D. M. J. and Duin, R. P. W. (1999). Support vector domain description. *Pattern Recognition Letters*, 20(11-13):1191–1199.

Yusoff, Y. A., Basori, A. H., and Mohamed, F. (2013). Interactive hand and arm gesture control for 2d medical image and 3d volumetric medical visualization. *Procedia-Social and Behavioral Sciences*, 97:723–729.

Zhu, X. and Goldberg, A. B. (2009). Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, 3(1):1–130.